НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
«ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ»
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

До захисту допущено
в. о. завідувача кафедри ММСА
О.Л.Тимощук
«___» _____ 2019 р.

На правах рукопису
УДК 519.254

Магістерська дисертація

на здобуття ступеня магістра за спеціальністю 124 Системний аналіз
на тему: «Методи інтелектуального аналізу даних для моделювання і прогнозування курсу криптовалют»

Виконав:
студент II курсу, групи КА-82 мп _____
Кінда Віталій Васильович
Науковий керівник:
_____
к.т.н, доц. Терентьєв О. М.
Рецензент:
к.ф-м.н, доцент кафедри прикладних
інформаційних систем Київського національного _____
університету імені Тараса Шевченка
Домрачев В. М.

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань
Студент _____

Київ
2019

РЕФЕРАТ

Магістерська дисертація: 95 с., 25 табл., 22 рис., 1 додаток, 31 джерел.

Актуальність теми: в світі бурхливо зростає новий ринок криптовалют. Проте, разом з цим, зростає і кількість трейдерів, основною задачею яких є одержання прибутку. Таким чином, розробка та застосування систем прогнозування курсу криптовалюти у процесі прийняття рішення щодо здійснення операцій купівлі продажу криптовалюти є актуальною на сьогоднішній день.

Мета даної роботи полягає у дослідженні та вдосконаленні існуючих методик побудови прогнозуючих моделей та розробці системи підтримки прийняття рішень для короткострокового прогнозування курсу криптовалют з використанням моделей експоненційного згладжування та нейроннних мереж.

Об’єктом дослідження є набір статистичних даних щодо операцій купівлі та продажу криптовалюти на біржі.

Методи дослідження: моделі експоненційного згладжування, нейроннні мережі та операції над матрицями.

Програмний продукт реалізований за допомогою мови програмування Python 3.7 у середовищі розробки Jupyter Notebook.

Отримані результати: розроблено систему підтримки прийняття рішень для короткострокового прогнозування курсу криптовалют з використанням моделей експоненційного згладжування та нейронних мереж. Проведено апробацію програмного продукту на реальних даних.

ФІНАНСОВИЙ РИНОК, КРИПТОВАЛЮТА, РЕКУРЕНТНІ НЕЙРОННІ МЕРЕЖІ, ПРОГНОЖУВАННЯ, СИСТЕМА ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ. ЗАГАЛЬНА ТОЧНІСТЬ МОДЕЛІ, ТРЕЙДИНГ, КРИПТОВАЛЮТНА БІРЖА.

ABSTRACT

Master's thesis explanatory note: 95 p., 25 tabl., 22 fig., 1 application, 31 references.

Topic: Data mining techniques for modeling and forecasting cryptocurrency exchange rates.

Relevance of the topic: a new cryptocurrency market is booming in the world. However, at the same time, there is an increasing number of traders whose main task is to make a profit.

Thus, the development and application of cryptocurrency exchange rate forecasting systems in the process of deciding whether to buy cryptocurrency sales transactions is relevant today.

The purpose of this work is to research and improve existing techniques for constructing forecasting models and developing a decision support system for short-term forecasting of cryptocurrencies using exponential smoothing models and neural networks.

The object of the study is a set of statistics on the operations of buying and selling cryptocurrency bitcoins on an exchange.

Research methods: exponential smoothing models, neural networks, and matrix operations.

The software is implemented using Python 3.7 programming language in the Jupyter Notebook development environment.

Results obtained: a decision support system for short-term forecasting of cryptocurrency rates using exponential smoothing models and neural networks has been developed. The software is tested on real data.

FINANCIAL MARKET, CRYPT CURRENCY, RECURRENT NEURAL NETWORKS, FORECASTING, DECISION SUPPORT SYSTEM. GENERAL PRECISION OF THE MODEL, TRADING, CRYPTOCURRENCY.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

БД – база даних

ММП – метод максимальної правдоподібності

МНК – метод найменших квадратів

ОПР – особа, яка приймає рішення

ПЕОМ – персональна електронно-обчислювальна машина

ПП – програмний продукт

СППР – система підтримки прийняття рішень

MAE (англ. Mean Absolute Error) – середня абсолютна похибка

MSE (англ. Mean Squared Error) – середньоквадратична похибка

MAPE (англ. Mean Absolute Percentage Error) - середня абсолютна відсоткова похибка

SES (англ. Simple Exponential Smoothing) - просте експоненційне згладжування

HES (англ. Holt’s Exponential Smoothing) - експоненційне згладжування Хольта

DES (англ. Damped Exponential Smoothing) - демпфуюче експоненційне згладжування

LSTM (англ. Long Short Term Memory) - довга короткострокова пам’ять

ВСТУП

В світі бурхливими темпами розвивається новий ринок криптовалют. Разом з цим, зростає кількість компаній та фізичних осіб, основною задачею яких є торгівля валютою з метою отримання прибутку. Таким чином, розробка та застосування систем прогнозування курсу криптовалют у процесі прийняття рішення щодо здійснення операцій купівлі продажу криптовалюти є актуальною на сьогоднішній день.

Мета даної роботи полягає у дослідженні та вдосконаленні існуючих методик побудови прогнозуючих моделей та розробці системи підтримки прийняття рішень для короткострокового прогнозування курсу криптовалют на основі інтелектуального аналізу даних з використанням моделей експоненційного згладжування та нейроннних мереж.

Об’єктом дослідження є набір фінансових даних щодо операцій купівлі та продажу криптовалюти на торговій онлайн-платформі.

Предмет дослідження – математичні методи побудови предиктивних моделей, а саме: інформаційна технологія, нейронні мережі довгої короткострокової пам’яті (LSTM) та моделі експоненційного згладжування.

Методи дослідження: підхід системного аналізу, методи системнго аналізу, методи інтелектуального аналізу даних, методи дослідження часових рядів, моделі експоненційного згладжування, нейроннні мережі LSTM архітектури.

У рамках дисертації необхідно вирішити такі задачі:

а) проаналізовати існуючі рішення для прогнозування фінансових часових рядів;
б) провести огляд та аналіз існуючих математичних методів моделювання і прогнозування криптовалютних котирувань;
в) розробити архітектуру системи підтримки прийняття рішень для аналізу, моделювання та прогнозування курсу криптовалюти;
г) розробити програмний продукт, в якому реалізувати роботу із фінансовими часовими рядами за допомогою нейронних мереж та моделей експоненційного згладжування;

д) апробувати програмний продукт на реальних даних та провести порівняльний аналіз із обгрунтованим вибором кращої моделі.

Інструменти розробки: технологія ізоляції середовищ розробки, середовище програмування високого рівня python 3.7.

Практичним результатом роботи являється розроблений програмний продукт, який дозволить робити короткостроковий прогноз фінансових часових рядів на основі методів інтелектуального аналізу даних.

Робота складається з 4 розділів. В першому розділі розглядаються поняття та сутність криптовалюти, технології блокчейн, проводиться огляд програмного забезпечення, призначеного для фінансових установ та трейдерів, наводяться переваги та недоліки. У другому розділі вивчаються підходи та методи для побудови предиктивних моделей та процес попереднього аналізу і обробки даних. У третьому розділі дисертації описується розроблена СППР, а також проводиться порівняння результатів роботи системи. Четвертий розділ присвячено розробленню стартап-проекту на основі створеного програмного продукту.

Інформаційну базу дослідження становлять праці вітчизняних та закордонних науковців, чинне законодавство, протоколи та стандарти, електронні ресурси.

РОЗДІЛ 1

ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Проблеми та перспективи технології криптовалют

Прискорені темпи розвитку ринку криптовалют та їх інтеграція в систему господарських, операційних, фінансових та інших процесів визначають необхідність комплексного вивчення даного явища. Особливої актуальності додає те, що на державному рівні в останні місяці активізувалися обговорення щодо перспектив легалізації ринку криптовалют і можливостей використання його інструментів у господарській діяльності економічних агентів. Незважаючи на часом полярні погляди і підходи, що сформувалися на даний момент серед експертів щодо вирішення даного питання, розвиток крипторинку проходить досить швидкими темпами незалежно від його регулювання.

Зараз важко уявити роль, яку відіграватимуть криптовалюти в економіці наступних десятиліть. Але її можна передбачити. Саме точність прогнозів визначає стратегію успіху та приносить дохід.

У певний момент гроші почали набувати цифрову форму. Вони набували її поступово, поєднуючись все більше із новими інформаційними технологіями, поки не виникла абсолютно нова віртуальна грошова сутність - криптовалюта. Класичні платіжні системи та криптовалюта різняться настільки сильно, що до сих пір багато людей не можуть зрозуміти сутність нових валют та їхню відмінність від звичних фіатних валют.

Дослідження криптовалют будемо розглядати на прикладі біткоіну, оскільки він є одним із найбільш популярних та розповсюджених в порівнянні із іншими, а принцип роботи суттєво не відрізняється від інших.

В 2008 році розробник програмного забезпечення Сатоши Накамото запропонував біткоін як систему електронних платежів, засновану на криптографічній базі. Ідея полягала в тому, щоб створити засіб обміну, незалежним від будь-якої центральної влади, який міг би передаватись в електронному

вигляді безпечним, перевіреним та незмінним способом.

Для того, щоб розібратись із складнощами навколо біткоіну необхідно розділити його на дві складові. З одного боку, є біткоін-токен – фрагмент коду, який являє собою цифрову власність (на зазок віртуального боргового зобов'язання). З іншого боку, є біткоін-протокол – розподілена мережа, що підтримує реєстр балансів біткоін-токена. Обидва вони мають назву "біткоін".

Система дозволяє відправляти платежі між користувачами без проходження через центральний орган, такий як банк або платіжний шлюз. Він створений і зберігається в електронному вигляді. Біткоіни не друкуються, як долари або євро - вони виробляються комп'ютерами по всьому світу з використанням вільного програмного забезпечення.

1.2 Відмінність криптовалют від традиційних

На сьогодні, більшість світових країн не регламентує роботу із новими іноваційними технологіями криптовалют, це ускладнює та сповільнює процес її розвитку. В нашому випадку необхідно розділяти саму технологію криптовалют та криптовалюту - як інструмент фінансового ринку. Подальші дослідження проводяться із криптовалютами, як інструментом фінансового ринку.

Біткоін може бути використаний для оплати в електронному вигляді, якщо обидві сторони готові до транзакції. У цьому сенсі це як звичайні долари, євро або єни, які також використовуються в цифровому вигляді [3].

Але він відрізняється від звичайних цифрових валют декількома важливими пунктами:

а) Децентралізація.

Найбільш важливою характеристикою криптовалют є те, що вони децентралізовані. Жодна установа не контролює мережу. Він підтримуєть-

ся групою програмістів-ентузіастів і управляється відкритою мережею спеціалізованих комп'ютерів, розкиданих по всьому світу. Це привертає окремих осіб і групи, яким незручно проводити операції із фінансами за допомогою банків або державних установ.

Біткоін вирішує «проблему подвійних витрат» електронних валют (в якій цифрові активи можуть бути легко зкопійовані і використані повторно) завдяки оригінальній комбінації криптографії та економічних стимулів. В електронних фіатних валютах цю функцію виконують банки, що дає їм контроль над традиційною системою. У випадку біткоінів - цілісність транзакцій підтримується розподіленою і відкритою мережею, якою ніхто не володіє.

б) Обмежена пропозиція.

Фіатні валюти (долари, євро і т. д.) мають необмежену пропозицію, тобто центральні банки можуть випускати будь-яку їхню кількість, і можуть намагатися маніпулювати вартістю валюти по відношенню до інших. В кінцевому випадку власники відповідної валюти несуть збитки через її знецінення.

У випадку біткоіну, поставки строго контролюються базовим алгоритмом. Невелика кількість нових біткоінів генерується щогодини, і буде продовжувати рости із меншою швидкістю, поки не буде досягнутий максимум 21 млн монет. Це робить біткоін більш привабливим як актив. Якщо попит зростає, а пропозиція залишається тією ж, то цінність зростає.

в) Анонімність.

В той час як відправники традиційних електронних платежів зазвичай ідентифікуються (для цілей перевірки, а також для дотримання вимог по боротьбі з відмиванням грошей та іншими актами законодавства), теоретично користувачі криптовалют працюють в умовах напіванонімності. Оскільки не існує центрального «валідатора», користувачам не потрібно ідентифікувати себе при здійсненні транзакцій. Коли відправляється запит транзакції, протокол перевіряє всі попередні транзакції, щоб підтвердити, що відправник має необхідну кількість валюти, а та-

кож права для їх відправки.

На практиці кожен користувач ідентифікується за адресою свого гаманця. Транзакції можуть, за певних зусиль, відслідковуватися таким чином. Також правоохоронні органи розробили методи ідентифікації користувачів, якщо це необхідно.

Крім того, згідно з чинним світовим законодавством більшість бірж зобов'язані проводити автентифікацію особистості своїх клієнтів, перш ніж їм дозволять купувати або продавати біткоіни, що спрощує відстеження використання біткоінів. Оскільки мережа прозора, хід конкретної транзакції видно всім.

Це робить біткоін не ідеальною валютою для злочинців, терористів, учасників організованих злочинних груп для відмивання фінансів.

г) Незмінність.

Криптовалютні транзакції не можна скасувати, на відміну від електронних фіатних транзакцій.

Це пов'язано з тим, що не існує центрального «судді», який міг би сказати «добре, поверніть гроші». Якщо транзакція записана в мережі, і якщо пройшов певний період часу, її неможливо змінити. Це означає, що будь-яка транзакція в мережі біткоінів не може бути підроблена.

д) Ділимість.

Найменша одиниця біткоіна називається Сатоши. Це одна стомільйонна частина біткоїна. Це дає змогу проводити мікротранзакції, що традиційні електронні гроші не можуть забезпечити.

е) Не схильність інфляції.

У криптовалют реалізований складний механізм запобігання інфляції. Зокрема в мережі біткоін інфляція запобігає кількома особливостями:

- обмежена емісія в 21 млн. монет, яка не може бути змінена;
- випуск нових монет відбувається строго раз в 10 хвилин;
- кожні 4 роки емісія монет скорочується вдвічі.

Аналогічні методи є в кожної криптовалютной мережі, що дозволяє заздалегідь передбачити, скільки монет буде існувати в певний період часу. Крім того, не існує контролюючого органу, який міг би прийняти

одноосібне рішення про збільшення емісії.

Наведені відмінності криптовалют від фіатних валют складають її переваги. До цього списку можна добавити низькі транзакційні витрати. Транзакції в криптовалютних системах проходять за принципом P2P, без участі центрального контролюючого органу. Скорочення витрат на обслуговування мережі дозволяє істотно зменшити комісію за перекази. На відміну від банківських та електронних платіжних систем, користувачі мають можливість самостійно встановлювати розмір комісії і навіть відправляти транзакції без неї [4].

Іноваційні цифрові криптовалюти за лічені роки набрали величезну популярність і перевершили за надійностю світові валюти. Думка експертів на рахунок криптовалют та блокчейну розділилося. Багато хто вважає їх технологією, здатною змінити світ, інші - вбачають масу недоліків, що перешкоджають їх впровадженню.

Вивчивши переваги криптовалют, однозначно можна сказати, що це прогресивна технологія, що має величезний потенціал для розвитку. Але незважаючи на всі переваги, як і будь-яка технологія, криптовалюта не позбавлена і ряду недоліків:

а) Висока волатильність.

Це одна з особливостей криптовалют, що перешкоджає її глобалізації. На сьогоднішній день, курс криптовалют дуже мінливий і протягом коротких інтервалів часу може змінюватися в широкому діапазоні.

Ймовірно, що первісна волатильність викликана новизною активу і в міру збільшення кількості користувачів курс криптовалют повинен стати більш стабільним.

б) Ризик злому.

Оскільки криптовалюти існують в цифровому вигляді, вони можуть стати вразливими до кібератак. Сервіси, що працюють з криптовалютами повинні мати високий рівень безпеки для запобігання крадіжок. Слід зазначити, що дана вразливість викликана не самими особливостями криптовалют, а безпекою зберігання ключів доступу.

Також криптовалютні мережі схильні до так званої атакі 51%, коли біль-

ша частина потужності мережі сконцентрована в руках однієї людини, вона має право самостійно приймати рішення про транзакції (особливість децентралізованої системи). Така ситуація загрожує, перешкоджанням транзакцій інших користувачів.

1.3 Особливості технології блокчейн

Термін Blockchain частково характеризує його завдання і призначення. «Block» - це «блок», «chain» - це «послідовність». Виходить, що Blockchain - це послідовність блоків, котрі упорядковані між собою.

Блоки являють собою дані про транзакції, угоди і контракти всередині системи, представлені в криптографічній, зашифрованій формі. Всі блоки збудовані в послідовність, тобто пов’язані між собою. Для запису нового блоку, необхідно послідовне зчитування інформації про старі блоки.

Однорангова, децентралізована мережа (англ. Peer-to-peer, P2P) - це комп’ютерна мережа, заснована на принципі рівноправності учасників. Часто в такій мережі відсутні виділені сервери, а кожен вузол (peer) є як клієнтом, так і виконує функції сервера. На відміну від архітектури клієнт-сервера, така організація дозволяє зберігати працездатність мережі при будь-якій кількості і будь-якому поєднанні доступних вузлів. Учасниками мережі є всі піри. Рис. 1.1-1.2 [26].

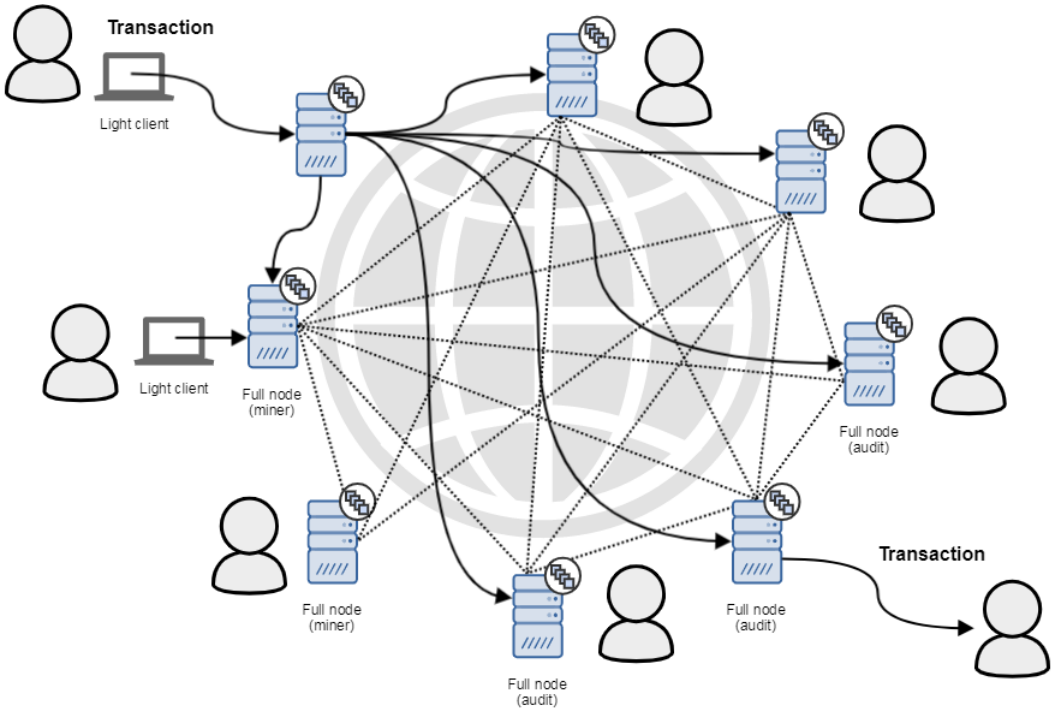

Рисунок 1.1 — Однорангова комп’ютерна мережа

Блокчейн складається з блоків транзакцій. Кожен блок містить заголовок і список транзакцій. У заголовку міститься хеш самого блоку, хеш попереднього блоку (тобто посилання на попередній блок), хеші всіх транзакцій і деяка службова інформація.

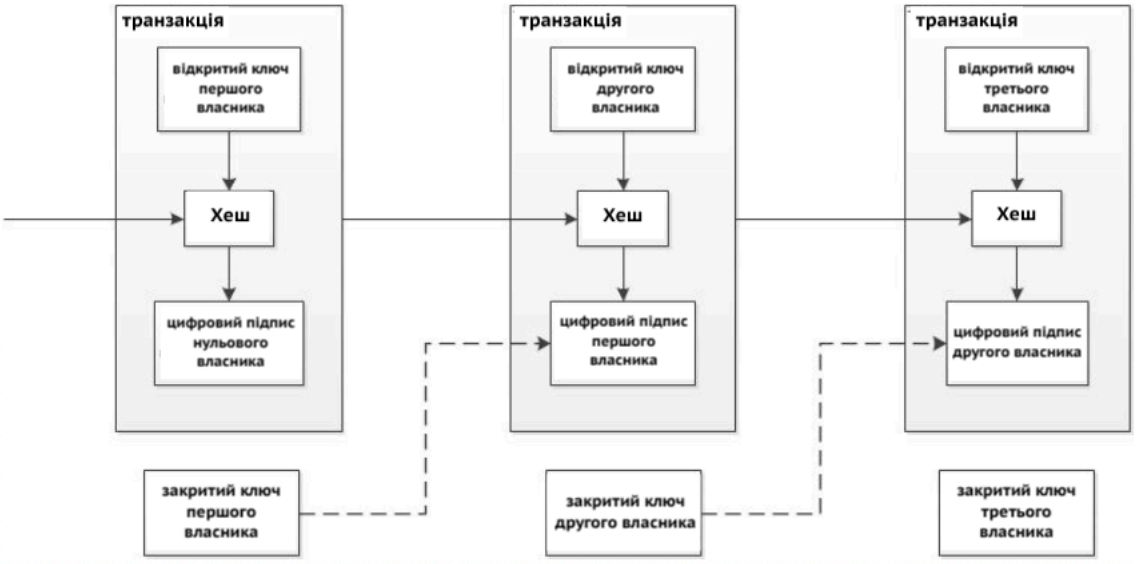

Рисунок 1.2 — Ланцюг блоків транзакцій

Хешування транзакцій проводиться за допомогою дерева Меркле. Для хешування використовується функція SHA-256 [28], яка вважається незворотною. Тому для підтвердження блоку транзакцій необхідно просто здійснювати підбір. Справа в тому, що в алгоритмі біткоіну є параметр - «складність». Це число, яке визначає значення хеш-функції, яку підбирає майнер для підтвердження блоку. Відповідно, блок приймається мережею тоді і тільки тоді, коли знаходиться такий хеш, який дорівнює або менше поточної складності в мережі. Рис 1.3 [27]. Складність змінюється динамічно так, щоб при зміні кількості задіяних в мережі потужностей для майнингу, швидкість майнингу лишалася незмінною - один блок в десять хвилин.

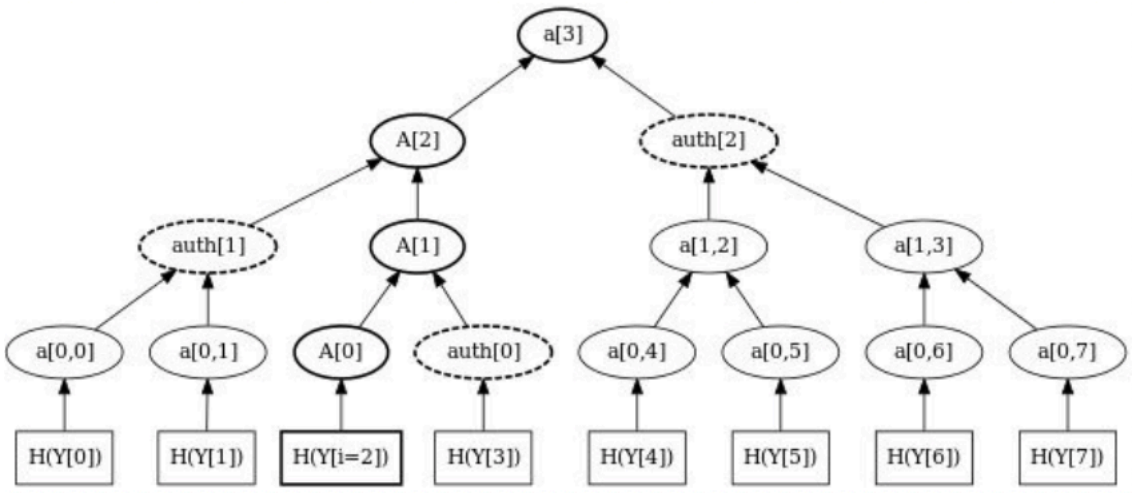

Рисунок 1.3 — Дерево Меркла

Застосування шифрування гарантує, що користувачі можуть змінювати тільки ті частини послідовності блоків, якими вони «володіють» в тому сенсі, що у них є закриті ключі, без яких запис в файл неможливий. Крім того, шифрування забезпечує синхронізацію копій розподіленого ланцюжка блоків у всіх користувачів.

Реалізується ще одна важлива функція: установка відносин довіри і підтвердження автентичності особистості, тому що ніхто не може змінювати ланцюжок блоків без відповідних ключів. Зміни, не підтверджені цими ключами, відхиляються. Звичайно, ключі (як і фізична валюта) теоретично мо-

жуть бути вкрадені, але захист кількох рядків комп'ютерного коду, зазвичай, не вимагає великих витрат.

Це означає, що основні функції, котрі виконуються банками (перевірка справжності особистості для запобігання шахрайства та подальша реєстрація угод, після чого вони стають законними), можуть виконуватися ланцюжком блоків швидше і точніше.

1.4 Огляд ринку програмного забезпечення призначеного для фінансового аналізу ринку криптовалют

Аналіз ринку програмного забезпечення та пакетів для бізнес аналітики можна провести за допомогою квадрантів Гартнера [29]. У своїх звітах Gartner розглядає не тільки якість і можливості програмного забезпечення, але і характеристики розробника в цілому, наприклад досвід продажів і роботи з клієнтами, повноту розуміння ринку, бізнес-модель, інновації, стратегії маркетингу, продажів, розвитку індустрії та ін. На основі оцінки по ключових параметрах претенденти розбиваються на 4 групи: лідери, претенденти на лідерство, далекоглядні та нішеві гравці.

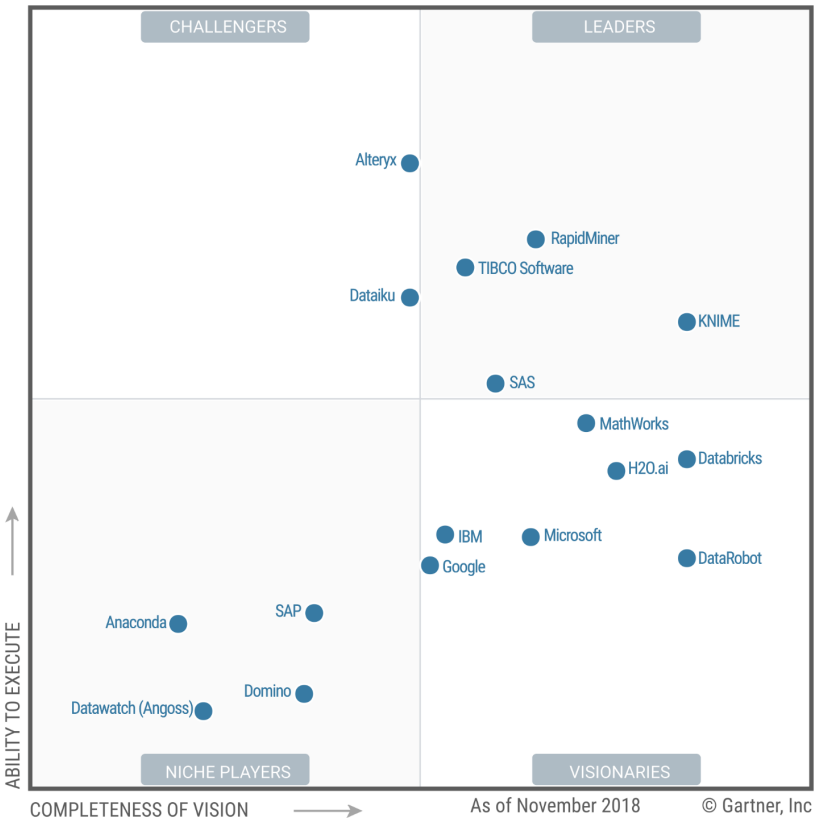

Рисунок 1.4 — Квадранти Гартнера для платформ машинного навчання

На рис. 1.4. приведено компанії, що займаються розробкою програмного забезпечення для машинного навчання поділеного на 4 групи.

Лідер інтелектуального аналізу даних та спеціалізованих рішень для прогнозування фінансових часових рядів є компанія Statistical Analysis System (SAS) [6]. Вона має широкий пакет спеціалізованих рішень для роботи з часовими рядами та інструментами для побудови регресійних моделей. Близько 50% банківського сектору США та 30% світового ринку в сфері бізнес-аналітики припадають саме на дану компанію. В рамках України - ТОП-30 найбільших банків користуються її інструментами [6]. Базовою мовою програмування в інфраструктурі SAS являється SAS base.

Реалізовані програмні системи аналізу даних, що представляють як статистичні методи, так і інтелектуальні методи. Розробники пропонують різноманітні методики аналізу, алгоритми оптимізації і інструменти створення власних програмних блоків. Досліднику, що проводить статистичну обробку даних, представлених часовими рядами, необхідне середовище розробки, що

відрізняється простотою і гнучкістю при виконанні основних завдань.

Мова програмування пакету статистичного аналізу даних SAS, яку відносять до мов четвертого покоління (4GL), орієнтована на виконання чотирьох основних завдань:

а) доступ до даних;
б) управління даними;
в) аналіз даних;
г) представлення даних.

З урахуванням специфічних особливостей пакету SAS, представляється доцільним використовувати дану систему при проведенні аналізу і прогнозуванні часових рядів, в яких рішення зазначених завдань аналізу і обробки даних займає центральне місце. Тим більше, що в складі системи SAS є пакет ETS (Econometrics and Time Series), що містить широкий спектр процедур статистичного аналізу часових рядів і налаштувань моделей прогнозування. На рис. 1.5 зображено вікно програмного продукту SAS [6].

Процедури, реалізовані в SAS/ETS, дозволяють застосовувати модель Бокса-Дженкінса (ARIMA), регресивні моделі з корекцією автокорельованих і гетероскедастичних процесів (AUTOREG), векторні авторегресійні моделі і моделі ковзного середнього (MA), проводити експоненційне згладжування (ESM), спектральний (SPECTRA) і фазовий (STATESPACE) аналіз часових рядів [7].

Таким чином, інтегрований набір пакетів SAS, відрізняючись винятковою простотою і гнучкістю при роботі з наборами даних, а також широким спектром доступних процедур статистичного аналізу, є одним з провідних інструментів аналізу та прогнозування часових рядів, поряд з іншими пакетами статистичного аналізу (SPSS, Minitab, R, STATA та ін.), інженерними пакетами (MatLab, Mathematica і ін.).

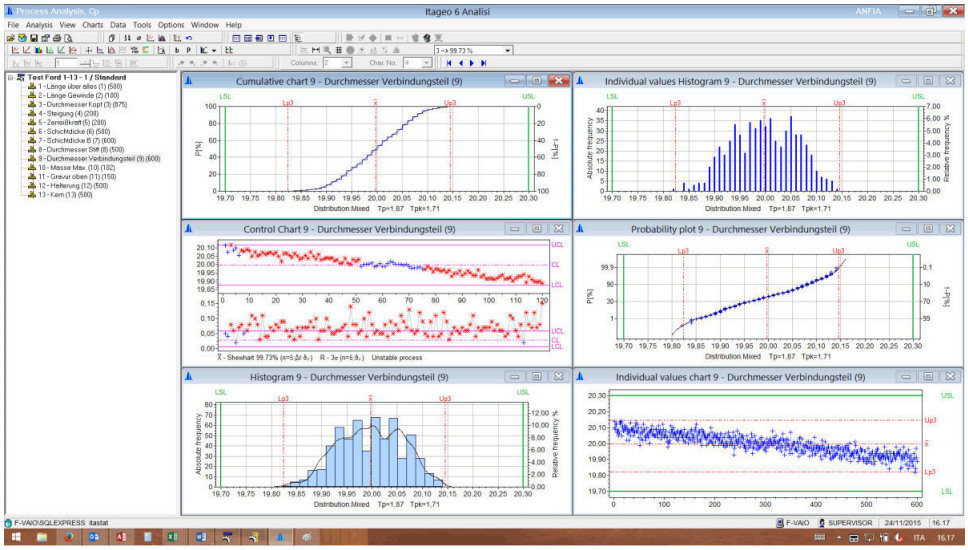

Рисунок 1.5 — Вигляд інтерфейсу SAS із пакетом ETS

Проте попри розвинуту системну інфраструктуру SAS, вона має ряд недоліків які суттєво послаблюють її позиції на ринку інтелектуального аналізу даних.

До них можна віднести наступне:

а) Відсутність безкоштовних демо-версій;

б) Висока вартість програмних рішень через щорічну підписку ліцензійного продукту;

в) Закритий код, що не дозволяє широкому колу користувачів вносити правки.

На відміну від пакетів продуктів компанії SAS, мною було обрано інфраструктуру Anaconda в зв’язці із Python в силу відкритості внутрішнього коду систем, простоти та гнучкості в користування, можливості безкоштовного використання. Дані перевагу зіграли основну роль у виборі платформи розробки.

1.5 Постановка задачі дослідження

Метою магістерської дисертації є дослідження та вдосконалення існуючих методів аналізу і короткострокового прогнозування фінансових часових рядів, розробка програмного забезпечення для інтелектуального аналізу даних, побудова моделей та перевірка їх на адекватність за обраними критеріями.

В рамках дисертації необхідно:

а) проаналізовати існуючі рішення для прогнозування фінансових часових рядів;
б) провести огляд та аналіз математичних методів моделювання і прогнозування криптовалютних котирувань для побудови на їх основі моделей;
в) розробити архітектуру системи підтримки прийняття рішень для аналізу, моделювання та прогнозування курсу криптовалюти;
г) розробити програмний продукт, в якому реалізувати роботу із фінансовими часовими рядами за допомогою нейронних мереж та моделей експоненційного згладжування;
д) апробувати програмний продукт на реальних даних та провести порівняльний аналіз із обгрунтованим вибором кращої моделі.

Висновки до розділу 1

Криптовалюти явили собою новю, іноваційну, віртуальну технологію, що набирає розповсюдження широкими темпами саме через її практичність та захищеність. З плином часу вони все більше будуть набирати популярність. Це дасть змогу перейти на новий рівень - рівень низьковолатильних валют.

На даний момент, маючи високу волатильність маємо змогу побудувати якомога точніші математичні моделі із використанням інтелектуального аналізу даних для прогнозування курсу криптовалют в короткостроковій перспективі.

Огляд ринку криптовалют та подальша робота із котируваннями та наборами даних буде проводиться на прикладі однієї криптовалюти - біткоін. Це обумовлено ціною, популярністю та широким застосуванням в порівнянні із іншими.

В даному розділі було розкрито сутність криптовалют, блокчейн технології, проведено огляд переваг та недоліків в порівнянні із фіатними валютами для ознайомлення із тематикою дослідження.

Проведено огляд програмного продукту, який є основними лідером на ринку програмного забезпечення призначеного для забезпечення поставленої задачі, а саме пакети рішень SAS.

Показано актуальність та перспективність дослідження, на основі чого сформульовано постановку задачі магістерської дисертації, та виділено етапи її розв’язку.

РОЗДІЛ 2

ОБГРУНТУВАННЯ МЕТОДИЧНИХ ПІДХОДІВ

2.1 Методика дослідження стаціонарності часових рядів

Стаціонарний процес - це стохастичний процес, у якого не змінюється розподіл ймовірності при зміщенні в часі. Під стаціонарністю розуміють властивість процесу не змінювати своїх статистичних характеристик з плином часу, а саме сталість математичного сподівання та сталість дисперсії (гомоскедастичність) і незалежність коваріаційної функції від часу (повинна залежати тільки від відстані між спостереженнями). Оскільки стаціонарність лежить в основі багатьох статистичних процедур, що використовуються в аналізі часових рядів, нестаціонарні процеси часто зводяться до стаціонарних.

Важливим видом нестаціонарного процесу, який не включає трендоподібну поведінку, є циклостаціонарний процес, який є стохастичним процесом, що циклічно змінюється з часом. Згідно роботи Шей Пелехі [16] можна виокремити наступне:

$(\xi_t, t \in T \subseteq R)$ - випадковий процес, визначений на імовірнісному просторі $(\Omega, \mathfrak{A}, \mathbb{P})$, називається 'стаціонарним в широкому сенсі', якщо $\forall t \in T \subseteq \mathbb{R}$ вірні такі властивості:

а) $\exists M\xi_t$ та $\exists D\xi_t \neq 0$;

б) Функція математичного сподівання постійна та не залежить від часу t;

в) Коваріаційна функція фукціонально залежить лише від різниці аргументів $cov(\xi_t, \xi_s) = K(t, s) = \hat{K}(t - s), \forall s \in T$.

Простим випадком стаціонарного процесу є білий шум.

Наочно можна подивитися на ці властивості на ілюстраціях, взятих з роботи Sean Abu Рис. 2.1 [22]:

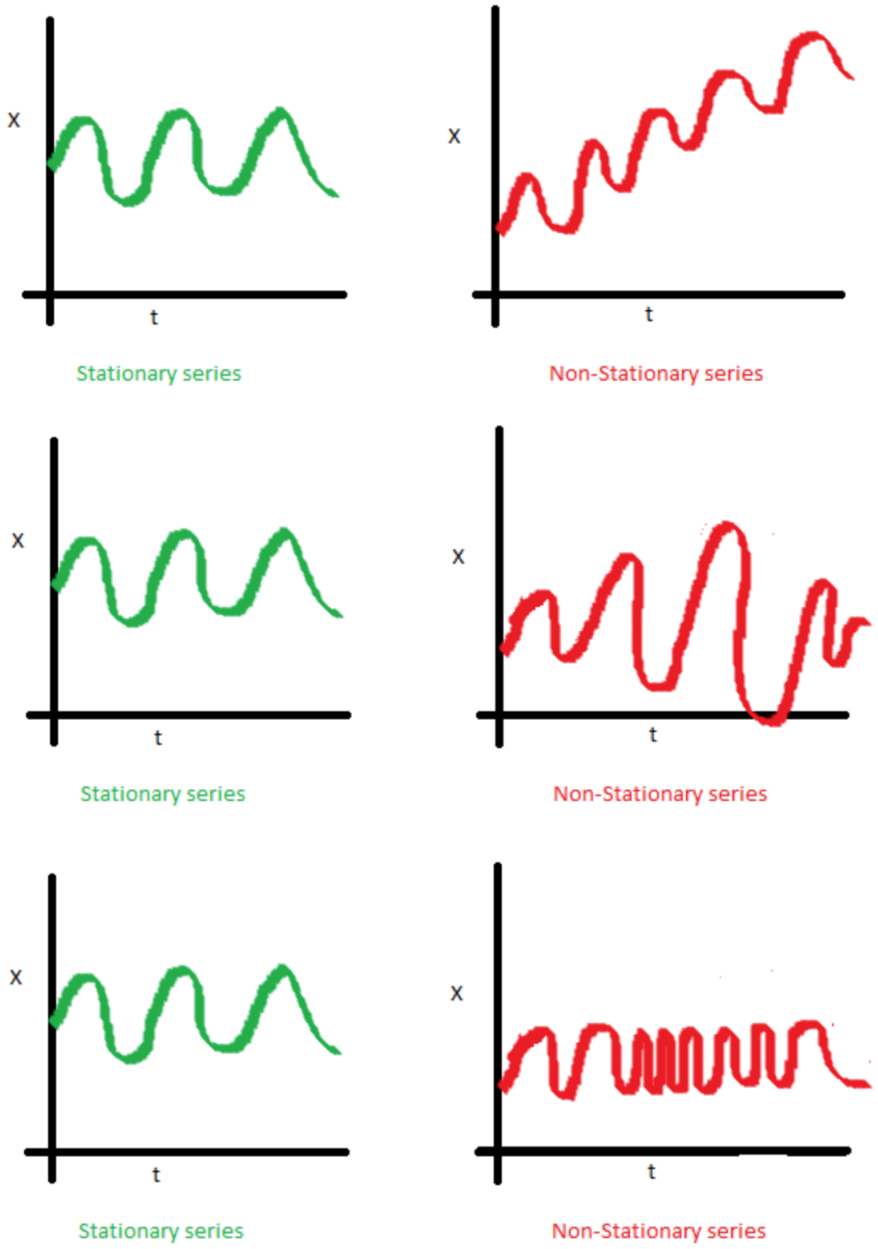

Рисунок 2.1 — Приклади стаціонарних(зліва) та нестаціонарних(зправа) процесів

Для першого випадку розкид значень ряду істотно варіюється в залежності від періоду. В другому випадку значення ряду раптово стають ближчими один до одного, утворюючи деякий кластер, а в результаті отримуємо мінливість коваріацій. В останньому випадку часовий ряд зправа не є стаціонарним, так як його матсподівання зростає.

Стаціонарність процесу є важливою, оскільки по ньому легко будувати

прогноз, так як ми вважаємо, що його майбутні статистичні характеристики не будуть відрізнятися від спостережуваних поточних. Більшість моделей часових рядів так чи інакше моделюють і прогнозують ці характеристики (наприклад, математичне сподівання або дисперсію), тому в разі нестаціонарності вихідного ряду прогноз виявиться невірними. На жаль, більшість часових рядів, з якими доводиться стикатися не є стаціонарними, але з цим можна та потрібно боротися.

2.1.1 Перевірка на стаціонарність. Розподіл Дікі-Фуллера

Ціну криптовалюти можна розглядати як авторегресійний процес першого порядку:

$$y_t = \phi y_{t-1} + \epsilon_t, \tag{2.1}$$

де ϕ - параметр моделі, $\epsilon_t \sim N(0, \sigma^2)$ - білий шум, $t = 1, \cdots, n$. Такий процес називається стаціонарним при умові $|\phi| < 1$.

Припустимо, у нас є певний набір даних із значенням курсу криптовалют за певний проміжок часу. Постає завдання за наявними спостереженнями визначити, чи є такий авторегресійний процес стаціонарним чи ні. Необхідно провести стандартну процедуру тестування гіпотези:

а) $H_0 : \phi = 1$ - тобто процес не стаціонарний;

б) $H_1 : |\phi| < 1$ - альтернативна гіпотеза (тобто процес стаціонарний).

Насправді, з тестуванням гіпотези не все так просто, тому що якщо справжнє значення $\phi = 1$, t-статистика не розподілена згідно із законом Стьюдента і її розподіл не прямує до стандартного нормального при збільшенні кількості спостережень. В такому випадку ми не можемо просто взяти таблицю критичних значень Стьюдента і перевірити по ній гіпотезу [11].

Під t-статистикою тут розуміється відношення відхилення оцінки параметра авторегресійної моделі від його істинного значення до стандартної

помилки оцінки коефіцієнта:

$$t = \frac{\hat{\phi} - \phi}{s_{\hat{\phi}}}, \tag{2.2}$$

де $\hat{\phi}$ - оцінка параметра авторегресії першого порядку, $s_{\hat{\phi}}$ - стандартна похибка оцінки $\hat{\phi}$. Оцінка коефіцієнта $\hat{\phi}$ в альтернативній моделі може будуватися за допомогою звичайного методу найменших квадратів (МНК).

В роботі [23] представлено розподіл t-статистики за умови $\phi = 1$, тобто при

$$t = \frac{\hat{\phi} - 1}{s_{\hat{\phi}}},$$

яке отримало назву статистики Дікі-Фуллера, для рівняння (2.1) і двох його модифікацій:

$$y_t = a + \phi y_{t-1} + \epsilon_t, \tag{2.3}$$

$$y_t = a + \phi y_{t-1} + ct + \epsilon_t. \tag{2.4}$$

Для рівняння (2.1) розподіл Дікі-Фуллера має наступний вигляд [23]:

$$\lim t_1 = \frac{W^2(1) - 1}{\sqrt{\int_0^1 W^2(s)ds}}, \tag{2.5}$$

де t_1 - t-статистика для процесу (2.1), $W(s)$ - стандартний вінерівский процес.

Критичні значення статистики Дікі-Фуллера наведені в книзі Фуллера «Introduction to Statistical Time Series». Таким чином, для перевірки авторегресійного процесу на стаціонарність необхідно використовувати стандартну

процедуру тестування гіпотези з тією відмінністю, що замість таблиці критичних значень для розподілу Стюдента необхідно використовувати таблицю критичних значень для розподілу Діккі-Фуллера.

Також важливо відзначити, що рівняння (2.1), (2.3) і (2.4) можна переписати в наступному вигляді [23]:

$$\begin{aligned} y_t - y_{t-1} &= \phi y_{t-1} - y_{t-1} + \epsilon_t, \\ \Delta y_t &= (\phi - 1) y_t + \epsilon_t, \\ \Delta y_t &= \beta y_t + \epsilon_t, \end{aligned} \tag{2.6}$$

$$\begin{aligned} y_t - y_{t-1} &= a + \phi y_{t-1} - y_{t-1} + \epsilon_t, \\ \Delta y_t &= a + (\phi - 1) y_t + \epsilon_t, \\ \Delta y_t &= a + \beta y_t + \epsilon_t, \end{aligned} \tag{2.7}$$

$$\begin{aligned} y_t - y_{t-1} &= a + \phi y_{t-1} - y_{t-1} + ct + \epsilon_t, \\ \Delta y_t &= a + (\phi - 1) y_t + ct + \epsilon_t, \\ \Delta y_t &= a + \beta y_t + ct + \epsilon_t, \end{aligned} \tag{2.8}$$

де $\Delta y_t = y_t - y_{t-1}$, а $\beta = \phi - 1$. Процеси (2.6), (2.7) і (2.8) можуть бути оцінені і протестовані при $\beta = 0$ аналогічно тестуванню гіпотези при $\phi = 1$. Отже, статистика Дікі-Фуллера дозволяє здійснювати перевірку на стаціонарність не тільки самого процесу, але і його різниць першого порядку.

2.1.2 Приклад перевірки тесту Дікі-Фуллера

Процес, породжений стандартним нормальним розподілом є стаціонарним, коливається навколо нуля (має нульове математичне сподівання) з відхиленням в 1. Приклад генерації стаціонарного часового процесу представлений на Рис. 2.2. Тепер на підставі нього згенеруємо новий процес, в якому кожне наступне значення буде залежати від попереднього: $y_t = \phi y_{t-1} + \epsilon_t$.

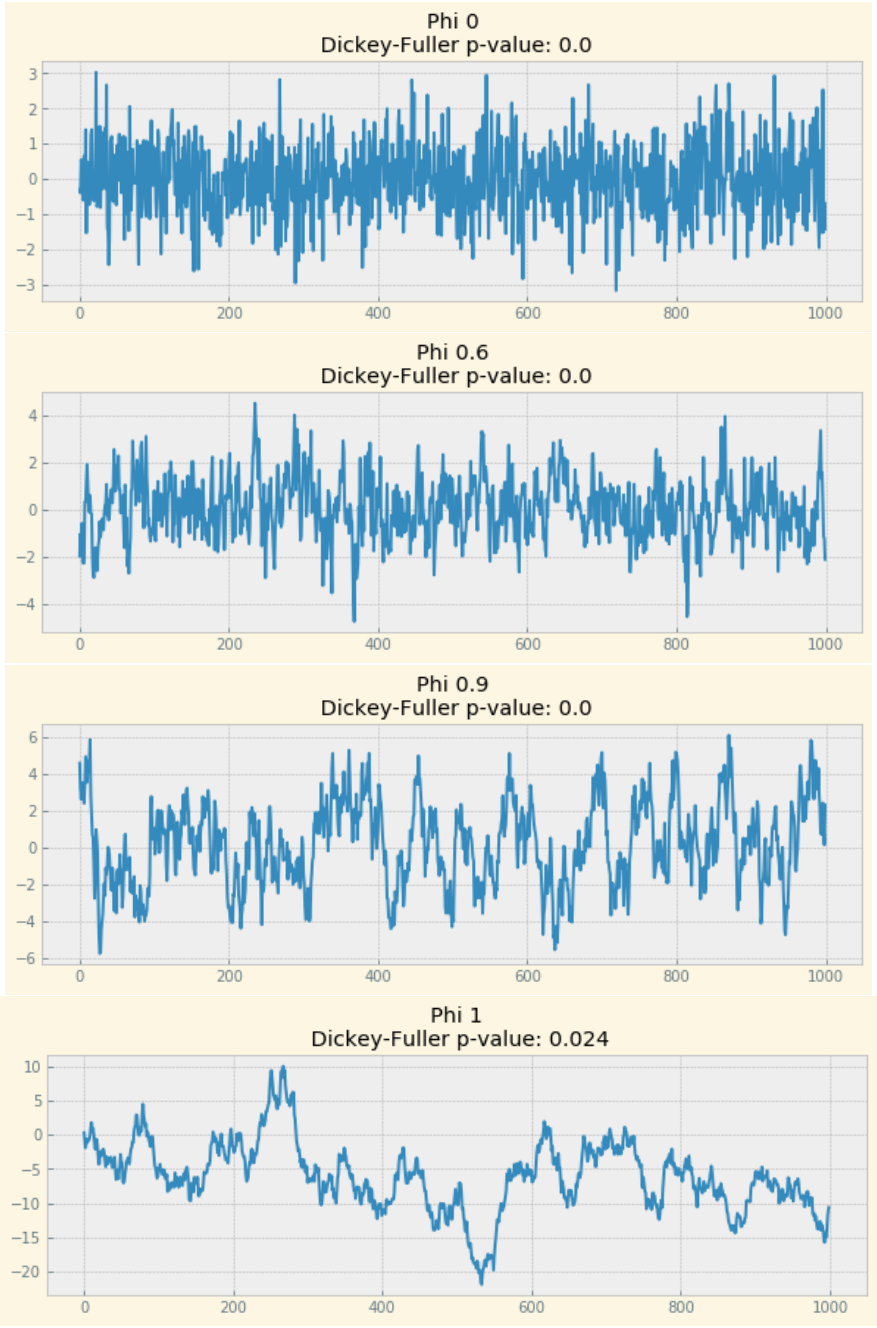

Рисунок 2.2 — Генерація нових процесів

На першому графіку вийшов такий самий стаціонарний білий шум, який будувався раніше. На другому, значення ϕ збільшилось до 0.6, в результаті чого на графіку стали з'являтися більш широкі цикли, але в цілому стаціонарним він бути не перестав. Третій графік все сильніше відхиляється від нульового середнього значення, але все ще коливається навколо нього. Нарешті, значення ϕ рівне одиниці дало процес випадкового блукання - ряд не

стаціонарний.

Відбувається це через те, що при досягненні критичної одиниці, ряд $y_t = \phi y_{t-1} + \epsilon_t$ перестає повертатися до свого середнього значення. Якщо відняти від лівої і правої частини y_{t-1}, то отримаємо $y_t - y_{t-1} = (\phi - 1)y_{t-1} + \epsilon_t$, де вираз зліва - перші різниці. якщо $\phi = 1$, то перші різниці дадуть стаціонарний білий шум ϵ_t. Цей факт ліг в основу тесту Дікі-Фуллера на стаціонарність ряду (наявність одиничного кореня). Якщо з нестаціонарного ряду першими різницями вдається отримати стаціонарний, то він називається інтегрованим першого порядку. Нульова гіпотеза тесту - ряд не стаціонарний, відкидалася на перших трьох графіках, і прийнялась на останньому. Варто сказати, що не завжди для отримання стаціонарного ряду вистачає перших різниць, так як процес може бути інтегрованим з більш високим порядком (мати кілька одиничних коренів), для перевірки таких випадків використовують розширений тест Дікі-Фуллера, перевіряючий відразу декілька лагів.

Боротися з нестаціонарністю можна рядом способів - різницями різного порядку, виділенням тренда і сезонності, згладжуваннями і перетвореннями, наприклад, Бокса-Кокса або логарифмуванням.

2.2 Методи і моделі для вирішення задачі прогнозування фінансових часових рядів

2.2.1 Просте експоненційне згладжування

Методи експоненційного згладжування привласнюють експоненціально спадні ваги для минулих спостережень. Чим пізніше буде отримано спостереження, тим більшу вагу буде присвоєно. Наприклад, розумно додати більшої ваги до спостережень минулого тижня, ніж до спостережень, що відбувались 3 тижні тому. У 1985 році Gardner [24] запропонував ”єдину” класифікацію методів експоненційного згладжування. Проста модель часо-

вого ряду має наступний вигляд вигляд:

$$X_t = b + \epsilon_t,$$

де b - константа, ϵ - випадкова помилка. Константа b відносно стабільна на кожному часовому інтервалі, але може повільно змінюватися з часом

Один з інтуїтивно ясних способів виділення b полягає в тому, щоб використовувати згладжування ковзним середнім, в якому останнім спостереженнями приписуються більші ваги, ніж передостаннім, передостаннім більші ваги, ніж перед-передостаннім і т.д. Просте експоненційне згладжування саме так і влаштовано. Тут більш старим спостереженнями приписуються експоненціально спадні ваги, при цьому, на відміну від змінного середнього, враховуються всі попередні спостереження ряду, а не ті, що потрапили до певного вікна. Точна формула простого експоненціального згладжування має наступний вигляд [24]:

$$S_t = \alpha X_t + (1 - \alpha) S_{t-1},$$

У випадку застосування формули рекурсивно, кожне нове згладжене значення (яке є також прогнозом) обчислюється як зважене середнє поточного спостереження і згладженого ряду. Очевидно, результат згладжування залежить від параметра α. Якщо він рівний 1, то попередні спостереження повністю ігноруються. Якщо ж його значення рівне 0, то ігноруються поточні спостереження. Значення між 0, 1 дають проміжні результати.

Емпіричні дослідження показали, що часто просте експоненційне згладжування дає досить точний прогноз [24].

Оцінювання кращого значення параметра α доцільно виконувати за допомогою даних. На практиці параметр згладжування часто шукається з пошуком на сітці. Можливі значення параметра розбиваються сіткою з певним кроком дискретизації. Наприклад, розглядається сітка значень від 0.01 до 0.99, з

кроком дискретизації 0.05. Потім вибирається, для якого сума квадратів (або середніх квадратів) залишків (спостережувані значення мінус прогнози на крок вперед) є мінімальною. Тобто обирається краща модель згідно обраної метрики.

2.2.2 Подвійне експоненційне згладжування

Просте експоненційне згладжування в кращому випадку дає прогноз лише на одну точку вперед та ще може згладити ряд. Цього, на жаль, недостатньо, тому переходимо до розширення експоненційного згладжування, яке дозволить будувати прогноз відразу на дві точки вперед.

У цьому нам допоможе розбиття ряду на дві складові - рівень (level, intercept) l і тренд (trend, slope) b. Рівень, або очікуване значення ряду, ми прогнозували за допомогою попередніх методів, а тепер таке ж експоненційне згладжування можна застосувати до тренду, вважаючи, що майбутній напрямок зміни ряду залежить від зважених попередніх змін [24].

$$l_x = \alpha y_x + (1 - \alpha)(l_{x-1} - b_{x-1}), \tag{2.9}$$

$$b_x = \beta(l_x - l_{x-1}) + (1 - \beta)b_{x-1}, \tag{2.10}$$

$$\hat{y}_{x+1} = l_x + b + x. \tag{2.11}$$

В результаті отримуємо набір функцій. Перша визначає рівень (2.9) - він, як і раніше, залежить від поточного значення ряду, а другий доданок тепер розбивається на попереднє значення рівня та тренда. Друга відповідає за тренд (2.10) - він залежить від зміни рівня на поточному кроці, і від попереднього значення тренду. Тут в ролі ваги в експоненційному згладжуванні

виступає коефіцієнт β. Нарешті, підсумкове прогнозування є сумою модельних значень рівня і тренду (2.11).

Тепер налаштовувати довелося вже два параметра - α і β. Перший відповідає за згладжування ряду навколо тренду, другий - за згладжування самого тренду. Чим вище значення, тим більшу вага буде віддаватися останніми спостереженнями і тим менш плавним виявиться модельний ряд. Комбінації параметрів можуть видавати суттєво різні результати, особливо якщо ставити їх навмання.

2.2.3 Потрійне експоненційне згладжування

Ідея цього методу полягає в додаванні ще однієї, третьої, компоненти - сезонності. Відповідно, метод можна застосовувати тільки в разі, якщо ряд цієї сезонності не обділений. Сезонна компонента в моделі буде пояснювати повторювані коливання навколо рівня і тренду, а характеризуватися вона буде довжиною сезону - періодом, після якого починаються повторення коливань. Для кожного спостереження в сезоні формується своя компонента, наприклад, якщо довжина сезону становить 7 (наприклад, тижнева сезонність), то отримаємо 7 сезонних компонент, по одній на кожен із днів тижня.

Отримуємо нову систему:

$$\begin{aligned} l_x &= \alpha(y_x - s_{x-L}) + (1-\alpha)(l_{x-1} + b_{x-1}), \\ b_x &= \beta(l_{x-1} + b_{x-1}) + (1-\beta)b_{x-1}, \\ s_x &= \gamma(y_x - l_x) + (1-\gamma)s_{x-Ls}, \\ \hat{y}_{x+m} &= l_x + mb_x + s_{x-L+1+(m-1)modL}. \end{aligned} \tag{2.12}$$

Рівень тепер залежить від поточного значення ряду за вирахуванням

відповідної сезонної компоненти, тренд залишається без змін, а сезонна компонента залежить від поточного значення ряду за вирахуванням рівня і від попереднього значення компоненти (2.12) [23]. При цьому компоненти згладжуються через всі доступні сезони, наприклад, якщо це компонента, що відповідає за понеділок, то і згладжуватись вона буде тільки з іншими понеділками. Тепер, маючи сезонну компоненту, ми можемо прогнозувати вже не на один, і навіть не на два, а на довільну кількість кроків вперед.

2.2.4 Моделі з урахуванням сезонності та тренду

Більш складні моделі, на відміну від простого експоненційного згладжування, враховують тренд та компоненту сезонності. Сутність яких полягає в тому, що прогнози обчислюються не тільки за попередніми спостереженнями (як в простому експоненційному згладжуванні), але і з деякими затримками, що дозволяє незалежно оцінити тренд і сезонну складову. У 1985 році Gardner обговорив різні моделі та провів їх класифікацію.

В термінах сезонності можна виділити:

а) відсутність сезонності;
б) аддитивна сезонність;
в) мультиплікативна.

В категорії тренду можна розділити на наступні групи:

а) відсутність тренду;
б) лінійний тренд;
в) експоненційний тренд;
г) демпфований.

Багато часових рядів мають сезонні компоненти. Наприклад, продажі новорічних прикрас мають особливий попит лише в зимовий сезон. Маємо лише зимовий пік, в інші пори року кількість продаж різко падає. Це доволі пряма та наглядна ілюстрація сезонності. Така періодичність має місце що-

року. Однак відносний розмір продажів може трохи змінюватися з року в рік. Таким чином, є сенс незалежно експоненційно згладити сезонну компоненту додатковим параметром, зазвичай позначається δ.

Сезонні компоненти, за своєю природою, можуть бути адитивними або мультиплікативними. Наприклад, протягом грудня продажі певного виду товару збільшуються на 1 мільйон доларів щороку. Для того щоб врахувати сезонне коливання, можна додати в прогноз на кожен грудень 1 мільйон доларів (понад відповідний річного середнього). В цьому випадку сезонність - адитивна. Альтернативно, нехай в грудні продажі збільшилися на 40%, тобто в 1.4 рази. Тоді, якщо загальні продажі малі, то абсолютне (в доларах) збільшення продажів в грудні теж відносно мале. Якщо ж в цілому продажі великі, то абсолютне (в доларах) збільшення продажів буде пропорційно більшим. Знову, в цьому випадку продажі збільшаться в певну кількість разів, і сезонність буде мультиплікативної (в даному випадку мультиплікативна сезонна складова була б рівна 1.4). Відмінність між двома видами сезонності полягає в тому, що в адитивної моделі сезонні флуктуації не залежать від значень ряду, тоді як в мультиплікативної моделі величина сезонних флуктуацій залежить від значень часового ряду.

Загалом, прогноз на один крок вперед обчислюється наступними формулами [25]:

Адитивна модель:

$$Pr_t = S_t + I_{t-p}. \tag{2.13}$$

Мультиплікативна модель:

$$Pr_t = S_t * I_{t-p}, \tag{2.14}$$

де S_t позначає (просте) експоненціно згладжене значення ряду в момент t, і I_{t-p} позначає згладжений сезонний фактор в момент $t - p$, p - довжина

сезону. Таким чином, в порівнянні з простим експоненційним згладжуванням, прогноз ”поліпшується”додаванням або множенням сезонної компоненти. Ця компонента оцінюється незалежно за допомогою простого експоненційного згладжування наступним чином (2.15) та (2.16) [25]:

Адитивна модель:

$$I_t = I_{t-p} + \delta * (1 - \alpha) * e_t. \tag{2.15}$$

Мультиплікативна модель:

$$I_t = I_{t-p} + \delta * (1 - \alpha) * e_t / S_t, \tag{2.16}$$

де сезонна компонента в момент t обчислюється, як відповідна компонента на останньому сезонному циклі плюс помилка (e_t - різниця спостережного та прогнозованого значень в момент t). Параметр δ приймає значення між 0 і 1. Якщо він рівний 0, то сезонна складова на наступному циклі та сама, що і на попередньому. Якщо ж він рівний 1, то сезонна складова ”максимально”змінюється на кожному кроці через відповідної помилки. У більшості випадків, коли сезонність присутня, оптимальне значення δ лежить між 0 і 1.

2.3 Нейронні мережі

2.3.1 Перцептрон

Нейронні мережі доволі добре вивчені та їх можна вважати старою конструкцією. Вперше математична модель нейронна з’явилась у статті Уорена

Маккалоха і Уолтера Пітса [1] у 1943 році. У ній нейрон – це логічна модель, і не має відношення до машинного навчання.

Фундаментальною, «неподільною» частиною нейронної мережі є перцептрон. Перша конструкція штучного нейрона була описана Френком Розенблатом [1] у 1950-х роках.

Перцептрон Розенблата – це лінійна модель класифікації. Вважатимемо, що на вхід подається вектор дійсних чисел $\boldsymbol{x} = (x_1, x_2, \ldots, x_d) \in \mathbb{R}^d$, а виходи $y(x) \in \{1, -1\}$, тобто маємо справу з бінарною класифікацією. «Лінійна модель» означає, що ми шукатимемо такі ваги $w_0, w_2, \ldots, w_d$, щоб знак

$$\text{sign}(w_0 + w_1 x_1 + \ldots w_d x_d)$$

якомога частіше збігався з правильною відповіддю $y(x)$. Для зручності запису введемо в вектор $\boldsymbol{x}$ фіктивну координату, яка завжди матиме значення 1.

Нейронна мережа повинна бути спроможною навчитись наближати дуже складні функції, але якщо виходи кожного перцептрона будуть лінійними, то лінійною також буде і вся послідовність з'єднаних нейронів. Для того, щоб вирішити цю проблему скористаємось *логістичним сигмоїдом* замість sign:

$$\sigma(\boldsymbol{x}) = \frac{1}{1 + e^{-\boldsymbol{x}}}.$$

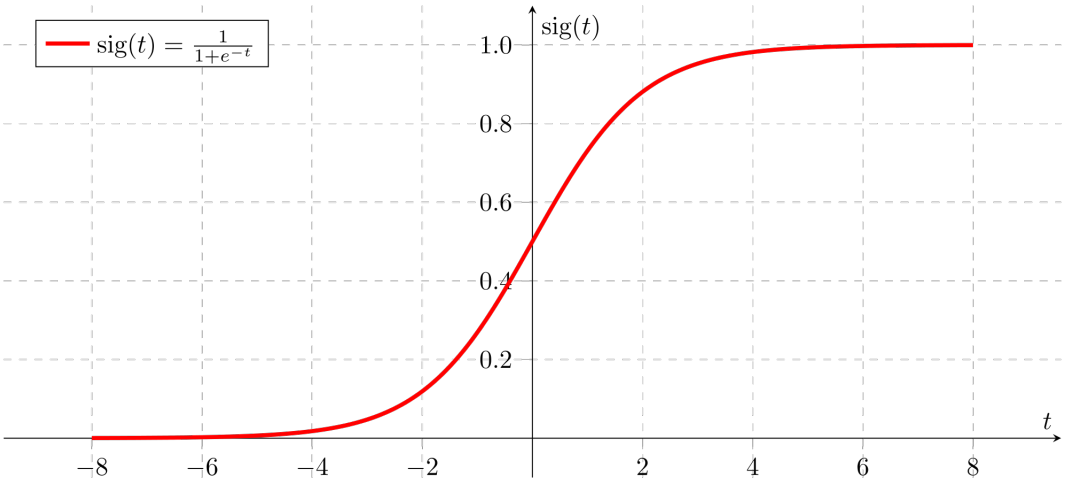

Рисунок 2.3 — Логістичний сигмоід

Логістичний сигмоід зображено на рис. 2.3. Такий перцептрон навчати не складно за допомогою градієнтного спуску, але тепер в якості функції помилки ми візьмемо пересічну ентропію [1]:

$$E(\boldsymbol{w}) = -\frac{1}{N}\sum_{i=1}^{N}\left(y_i \log \sigma(\boldsymbol{w}^T\boldsymbol{x}_i) + (1-y_i)\log\left(1-\sigma(\boldsymbol{w}^T\boldsymbol{x}_i)\right)\right).$$

Від цієї функції нескладно взяти похідну. Таким чином, один перцептрон з сигмоідом в кінці реалізує логістичну регресію і лінійно розділяє приклади.

2.3.2 Функції активації

Ми визначили сигмоіду як функцію, що допомагає створити нелінійність на виході перцептрона, але це не єдина можлива функцій і сучасних нейронних мережах часто надають перевагу іншим функціям. Логістичний сигмоід

дуже зручний оскільки добре обмежений та легко диференціюється, але це не єдина функція з такими властивостями.

Гіперболічний тангенс:

$$\tanh(\boldsymbol{x}) = \frac{e^{\boldsymbol{x}} - e^{-\boldsymbol{x}}}{e^{\boldsymbol{x}} + e^{-\boldsymbol{x}}}.$$

Можна вважати, що і гіперболічний тангенс це просто зміщений і розтягнутий сигмоід. Його похідна, виражається через самого себе

$$\tanh'(\boldsymbol{x}) = 1 - \tanh^2(\boldsymbol{x}).$$

Однією з найпопулярніших сьогодні функцій активації є ReLU (rectified linear units):

$$\mathrm{ReLU}(\boldsymbol{x}) = \begin{cases} 0, & \text{якщо } \boldsymbol{x} < 0, \\ \boldsymbol{x}, & \text{якщо } \boldsymbol{x} \geq 0. \end{cases}$$

Штучні нейрони з цією функцією активації використовувались ще у 1980-х в моделі багаторівневих мереж Куніхіко Фукусіми[30] для розпізнавання образів, які отримали назву Neocognitron.

ReLU-нейрони ефективніші, ніж ті, що базуються на логістичному сигмоіді чи гіперболічному тангенсі. Наприклад, щоб порахувати $\sigma'(x)$ необхідно рахувати складну функцію $\sigma(x)$, а потім ще множити на $1 - \sigma(x)$. А щоб порахувати $\mathrm{ReLU}'(x)$, потрібно тільки одне порівняння: якщо x менше нуля, то повернути 0, якщо більше, — 1. Це дозволяє витрачати значно менше ресурсів на етапі навчання нейронної мережі. Формально, похідна ReLU не визначена в нулі, але на практиці це не є суттєвим [1].

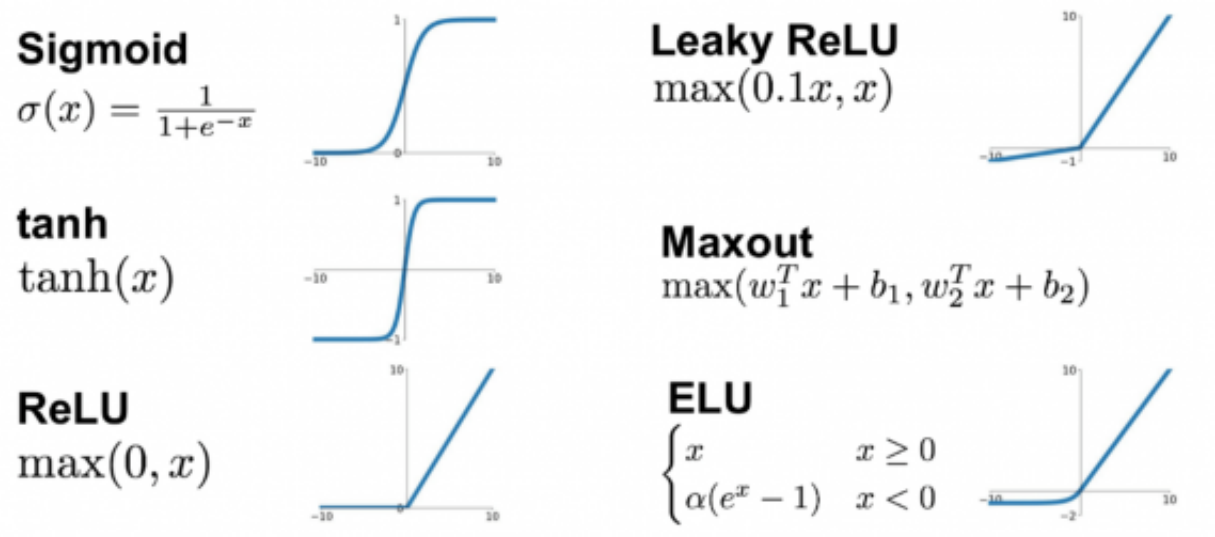

Рисунок 2.4 — Функції активації

Функції активації для нейронних мереж зображені на рис. 2.4 [1].

2.3.3 Long Short Term Memory мережа

У звичайних рекурентних мереж є недолік — у них «коротка пам’ять». Вплив прихованого стану експоненційно затухає. Рішення цієї проблеми полягає в тому, щоб ускладнити архітектуру однієї “цеглинки” рекурентної нейронної мережі — замість того, щоб мати одне число, на яке мають вплив усі подальші стани, ми можемо сконструювати комірку, в якій ми явно зможемо змоделювати “довгу пам’ять”.

Одне з найбільш популярних таких рішень — це LSTM (Long Short Term Memory). Стандартна архітектура LSTM комірки зображена; вона складається з трьох вузлів: вхідний, той, що забуває (forget) і вихідний рис. 2.5 [1].

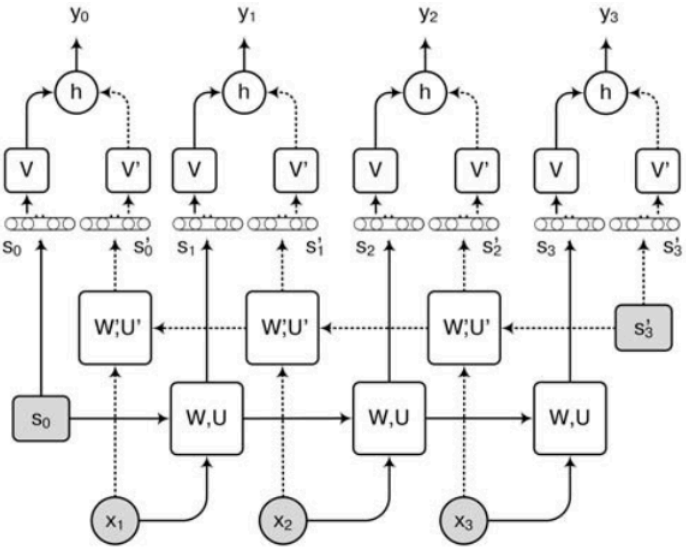

Рисунок 2.5 — Двонаправлена LSTM нейронна мережа

Якщо через x_t позначити вхідний вектор в момент часу t, через h_t — вектор прихованого стану в t, через W_x — матриці коефіцієнтів, що застосовуються до входу, через W_h — матриці коефіцієнтів у рекурентних з'єднаннях, а через b — вектори вільних членів, то отримаємо такий формальний опис LSTM комірки [1]:

$$
\begin{aligned}
c'_t &= \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_{c'}),\\
i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i),\\
f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f),\\
o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o),\\
c_t &= f_t \odot c_{t-1} + i_t \odot c'_t,\\
h_t &= o_t \odot \tanh(c_t).
\end{aligned}
$$

На вхід LSTM комірки подаються два вектори: новий вектор вхідних даних x_t і вектор прихованого стану h_{t-1}. Окрім цього у кожної LSTM є "комірка пам'яті" — вектор, що виконує роль пам'яті. c'_t — це лише кандидат на нове значення пам'яті. Перед тим як записати його на місце c_{t-1}, значення-кандидат і старе значення проходять через ще два "гейти": вхідний i_t і забу-

ваючий f_t:

$$c_t = f_t \odot c_{t-1} + i_t \odot c'_t$$

Архітектура LSTM дозволяє вирішити проблему зникаючих градієнтів, яка заважала звичайним рекурентним мережам тренувати довгострокові залежності. Для того, щоб це зрозуміти потрібно уявити LSTM без забуваючого гейта, тобто $f_t = 1$ для усіх t. Тоді вектор пам'яті буде рахуватись так:

$$c_t = c_{t-1} + i_t \odot c'_t$$

Помилки в мережі з LSTM комірок пропагуються без змін і приховані стани LSTM можуть, якщо сама комірка не вирішить їх перезаписати, зберігати свої значення необмежено довго. Це вирішує проблему зникаючих градієнтів: незалежно від матриці рекурентних коефіцієнтів помилка сама собою не зменшуватиметься.

Нескладно помітити, що звичайну рекурентну нейронну мережу можна розглядати як частковий випадок LSTM, де деякі значення зафіксовані у вигляді констант. Якщо встановити забуваючий гейт f_t завжди рівним нулю, а в вхідному гейті i_t і вихідному гейті o_t в одиниці, то вийде, що попереднє значення комірки c_{t-1} просто забувається.

2.4 Вибір метрики

Однією із найпоширеніших критерії оцінки якості моделі в задачах прогнозування можна виділити середню абсолютну похибку (MAE - Mean Absolute Error) та середню квадратичну похибку (MSE - Mean Squared Error) [31].

Наведемо приклад їхнього розрахунку.

$$MSE = \frac{1}{n}\sum_{t=1}^{n}(y_t - \widehat{y}_t)^2,$$

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|y_t - \widehat{y}_t|,$$

де y_t - реальне значення, а $\widehat{y}_t$ - спрогнозоване значення, n - розмір вибірки.

Середньоквадратична похибка в порівнянні із середньоабсолютною є більш чутливішою до великих вибірок. А значить вона є і більш чутливою до викидів.

Для прикладу можна навести наступний варіант. Значення MSE рівне 100 є неадкватною характеристикою моделі, цільова змінна якої лежить в діапазоні 0, 1 та на малій вибірці. Але вона є адекватним результатом при знаходженні цільової змінної в діапазоні -10000, 10000.

Середня абсолютна відсоткова похибка (MAPE - Mean Average Percenage Error) являється показником точності прогнозування моделі. Вона виражає точність в відсотках та визначається за формулою

$$MAPE = \frac{100\%}{n}\sum_{t=1}^{n}\left|\frac{y_t - \widehat{y}_t}{y_t}\right|,$$

де y_t - реальне значення, а $\widehat{y}_t$ - спрогнозоване значення, n - розмір вибірки.

Множення на 100 % робить його відсотковою похибкою.

Концепція MAPE є доволі простою, проте має декілька серйозних недоліків:

– його не можна застосовувати, якщо серед вибірки є нульові значення;

– немає верхньої межі оцінки якості моделі, адже це залежить від поставленої задачі.

Перевагою MAPE є проста та логічна інтерпритованість результату. MAPE є інваріантним стосовно маштабу, що досить добре для даних із великою дисперсією, на відміну від критерію MSE.

Також можна ввести власну метрику яка характеризує те, чи правильно прогнозується напрямок тренду. Тренд може мати низхідний або висхідний

варіант (Up and Down). Тобто є 2 варіанти напрямку тренду реального часового ряду та 2 варіанти для прогнозованого.

Тобто маємо можливість побудувати confusion matrix для прогнозованого часового ряду та оцінювати якість побудованої моделі, для прикладу, метрикою f1 score.

Для визначення метрики $F1$ необхідно ввести допоміжні формули:

$$PPV = \frac{TP}{TP + FP}; \tag{2.17}$$

$$TPR = \frac{TP}{TP + FN}. \tag{2.18}$$

Тоді метрика $F1$ має наступний вигляд:

$$F_1 = 2 \times \frac{PPV \times TPR}{PPV + TPR}. \tag{2.19}$$

Висновки до розділу 2

В рамках реалізації даного розділу було проведено огляд існуючих математичних методів прогнозування для короткострокового прогнозування фінансових часових рядів.

Розглянуто основні принципи роботи з часовими рядами, досліджено стаціонарність часових рядів та обргрунтовано необхідність зведення нестаціонарних рядів до стаціонарних для побудови адекватної прогнозуючої моделі. Наведено приклад модифікації часового ряду із збереженням його стаціонарності та описано тест Дікі-Фуллера.

Описано методи які будуть використовуватися для прогнозування часових рядів, а саме - просте експоненційне згладжування, подвійне та потрійне експоненційні згладжування з урахуванням сезонності та тренду а також нейронну мережу LSTM.

Наведено ряд метрик для оцінювання якості прогнозу, а саме MSE (Mean Squared Error), MAE (Mean Absolute Error), MAPE (Mean Average Percentage Error) та власну, суть якої полягає в правильності прогнозування тренду та реальних даних. Серед них обрано дві базисні та обгрунтовано доцільність їхнього використання.

РОЗДІЛ 3

СИСТЕМА ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ДЛЯ ПРОГНОЗУВАННЯ КУРСУ КРИТПТОВАЛЮТ

3.1 Аналіз архітектури системи

В даному розділі описується СППР для прогнозування курсу криптовалют із застосуванням інтелектуального аналізу даних. На рисунку 3.1 наведена структура розробленої СППР. Варто зауважити, що структура реалізованої СППР представляє собою комплекс засобів для аналізу та обробки даних (маніпуляція із даними). Це дає змогу особі, що відповідальна за прийняття рішень скоротити час здійснення процесу прийняття рішення а також збільшити точність [32].

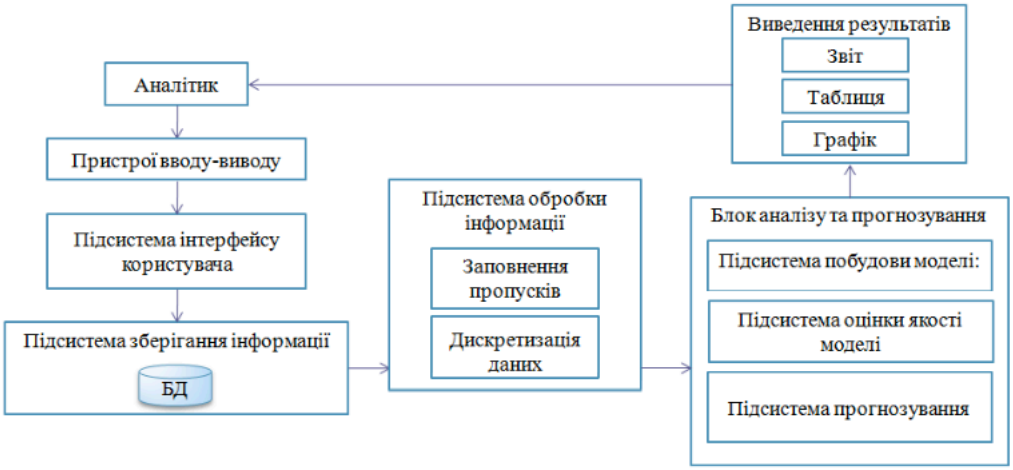

Рисунок 3.1 — Структура реалізованої СППР

Блок із пристроїв введення та виведення інформації дає можливість завантажувати дані та зберігати звіти в СППР. Тому підсистема вводу-виводу має тісний зв’язок із підсистемою інтерфейсу.

Блок зберігання інформації реалізована у вигляді сховища даних, яке призначена для роботи програмного продукту, з метою їх подальшого застосування.

Блок аналізу містить в собі три підсистеми, а саме: підсистема побудови ряду необхідних моделей, підсистема прогнозування та підсистема оцінювання якості моделі згідно заданого критерію.

Підсистема обробки інформації використовується для попередньої перевірки даних на відповідність необхідному стандарту.

Підсистема побудови моделі реалізована за допомогою нейронних LSTM мереж та ряду моделей експоненційного згладжування.

Підсистема оцінювання якості моделі обчислює значення критеріїм для ряду реалізованих моделей та проводить обрання кращої.

Підсистема виведення результатів містить реалізацію виведення необхідної інформації у вигляді зведених таблиць та звіту для прийняття рішення експертом.

Для створення СППР застосовувались технології Python 3.7 та середовище розробки Jupyter Notebook.

3.2 Основні технічні вимоги для коректної роботи програми

Для роботи програмного продукту необхідна наявність персонального комп’ютера з наступними мінімальними характеристиками:

а) ОС (операційна система комп’ютера) OSX Sierra;
б) мінімально допустима тактова частота процесора 2.4 ГГц;
в) об’єм оперативної пам’яті розміром 2048 Мбайт;
г) вільний дисковий простір: 10Мбайт;
д) пристрої вводу інформації — клавіатура та комп’ютерна ”мишка”;
е) пристрій виводу інформації — монітор з роздільною здатністю 2000×1024;
ж) інсталяція python 3.7 та пов’язаних бібліотек, jupyter notebook.

3.3 Попередній аналіз і обробка даних

3.3.1 Збір даних

Одним із найважливіших етапів для побудови якісної моделі є збір достатньої кількості вибірки даних фінансової історії криптовалютної біржі, тобто наявної інформації про купівлю чи продаж криптовалюти. Точність прогнозу та успіх розроблених моделей залежить від якості вхідних даних та процесу подальшої роботи із ними. На рис.3.2 показано приклад вікна торгової платформи.

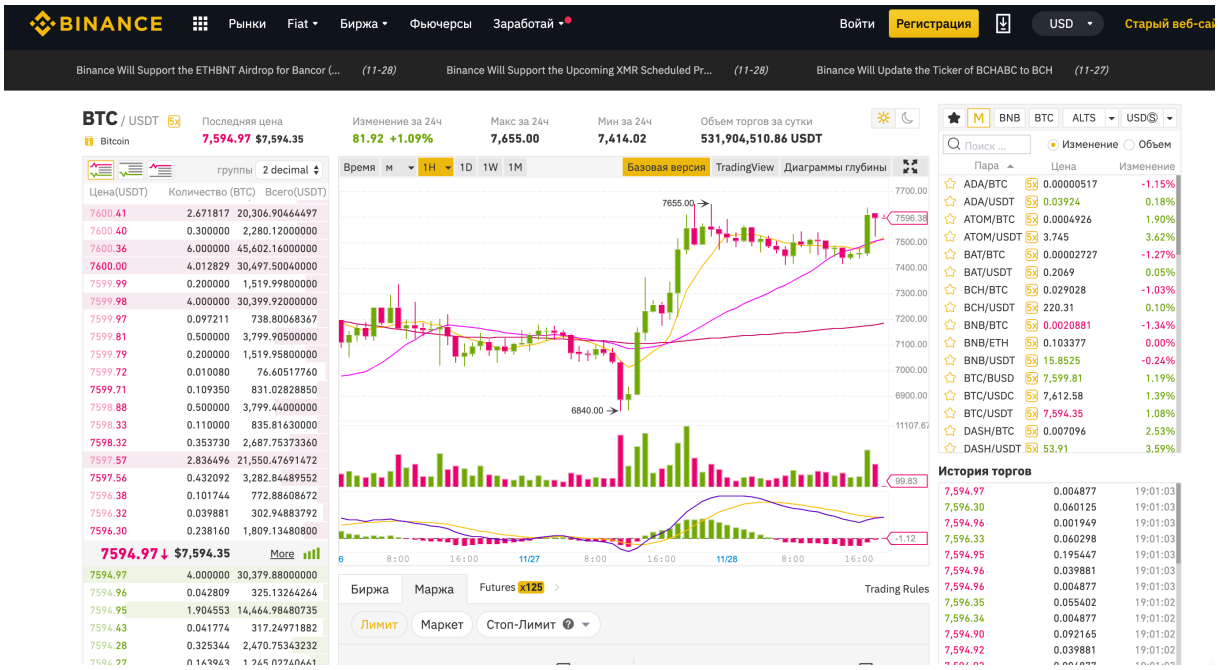

Рисунок 3.2 — Приклад вікна торгової платформи

Для побудови предиктивних моделей необхідно виконати аналіз даних, визначити цільову (таргетингову) та залежні змінні, перевірити дані на пропуски та здійснити набір інших, складніших маніпуляцій із даними.

Дані для моделювання були взяті із торгового майданчику poloniex.com за період від 2018-01-01 до 2019-01-01.

	Timestamp	Amount	Price
329	2018-01-01 02:00:00	0.014755	13769.0
328	2018-01-01 02:00:01	-0.100000	-13763.0
324	2018-01-01 02:00:02	0.010000	13766.0
327	2018-01-01 02:00:02	0.105929	13767.0
325	2018-01-01 02:00:02	0.025234	13766.0
326	2018-01-01 02:00:02	0.244811	13766.0
320	2018-01-01 02:00:03	0.300000	13765.0
323	2018-01-01 02:00:03	0.146274	13765.0
321	2018-01-01 02:00:03	0.015056	13765.0
322	2018-01-01 02:00:03	0.203304	13765.0
316	2018-01-01 02:00:04	-0.002196	-13763.0

Рисунок 3.3 — Загальний вигляд набору даних.

Основні змінні, що були завантажені з серверу мають наступний опис:

а) Timestamp - час здійснення операції.

б) Amount - величина, кількість криптовалюти, що приймає участь в транзакції. Якщо значення менше 0 - це означає операцію продажі, інакше - операцію купівлі.

в) Price - ціна транзакції (величина вимірюється в доларах США).

Після процесу feature engineering маємо нові згенеровані змінні наступного вигляду:

Timestamp	Min Buy price	Max Buy price	Mean Buy price	Min Sell price	Max Sell price	Mean Sell price	Sum Buy Amount	Sum Sell Amount	Total Count	Order Buy Count	Order Sell Count
2018-01-01 02:00:00	13508.000000	13788.000000	13610.928711	-13775.000000	-13505.000000	-13604.668945	613.295410	-837.185669	6527.0	2658.0	3869.0
2018-01-01 03:00:00	13252.000000	13655.000000	13473.172852	-13645.000000	-13251.000000	-13456.093750	414.783752	-995.882935	5130.0	1811.0	3319.0
2018-01-01 04:00:00	13222.000000	13440.000000	13337.716797	-13431.000000	-13214.000000	-13329.906250	574.535217	-652.944092	4383.0	2020.0	2363.0
2018-01-01 05:00:00	13322.000000	13591.000000	13448.921875	-13590.375000	-13310.000000	-13447.022461	470.243805	-594.678833	5180.0	2299.0	2881.0
2018-01-01 06:00:00	13266.000000	13583.000000	13422.141602	-13584.000000	-13262.000000	-13430.661133	412.749023	-442.447632	4570.0	2125.0	2445.0
...	...	...	...	...	...	...	...	...	...	...	...
2019-05-02 19:00:00	5719.899902	5779.799805	5748.965332	-5779.600098	-5719.799805	-5739.673340	91.832748	-182.868195	1467.0	588.0	879.0
2019-05-02 20:00:00	5723.100098	5750.000000	5740.774902	-5749.700195	-5723.000000	-5740.401367	68.878860	-31.279915	750.0	359.0	391.0
2019-05-02 21:00:00	5735.299805	5765.000000	5746.547363	-5760.700195	-5734.200195	-5743.663574	101.327667	-58.026207	860.0	410.0	450.0
2019-05-02 22:00:00	5736.985840	5755.799805	5746.853516	-5755.700195	-5736.700195	-5746.399414	80.283890	-57.436432	760.0	358.0	402.0
2019-05-02 23:00:00	5744.700195	5757.500000	5751.580566	-5757.399902	-5744.600098	-5751.304199	71.447212	-87.497162	749.0	269.0	480.0

Рисунок 3.4 — Загальний вигляд набору даних після feature engineering.

Основні змінні, що були згенеровані мають наступний опис:

а) Timestamp - час здійснення операції.
б) Min Buy price - мінімальна ціна серед транзакцій покупки;
в) Max Buy price - максимальна ціна серед транзакцій покупки;
г) Mean Buy price - середня ціна серед транзакцій покупки;
д) Min Sell price - мінімальна ціна серед транзакцій продажі;
е) Max Sell price - максимальна ціна серед транзакцій продажі;
ж) Mean Sell price - середня ціна серед транзакцій продажі;
з) Sum Buy Amount - загальна кількість купленої криптовалюти;
и) Sum Sell Amount - загальна кількість проданої криптовалюти;
к) Total Count - загальна кількість транзакцій;
л) Order Buy Count - загальна кількість транзакцій купівлі;
м) Order Sell Count - загальна кількість транзакцій продажі.

3.3.2 Візуалізація даних, перевірка на стаціонарність

Перш ніж будувати прогнозні моделі необхідно виконати крок data visualisation. Це дасть змогу аналітику ознайомитися з набором даних, з яким необхідно працювати. рис. 3.5.

Рисунок 3.5 — Графік зміни курсу валютної пари btc/usd.

Приклад виконання тесту Діккі Фуллера для перевірки часового ряду на стаціонарність. На рис. 3.6 наведено використання тесту до та після модифікації часового ряду а також АКФ та ЧАКФ.

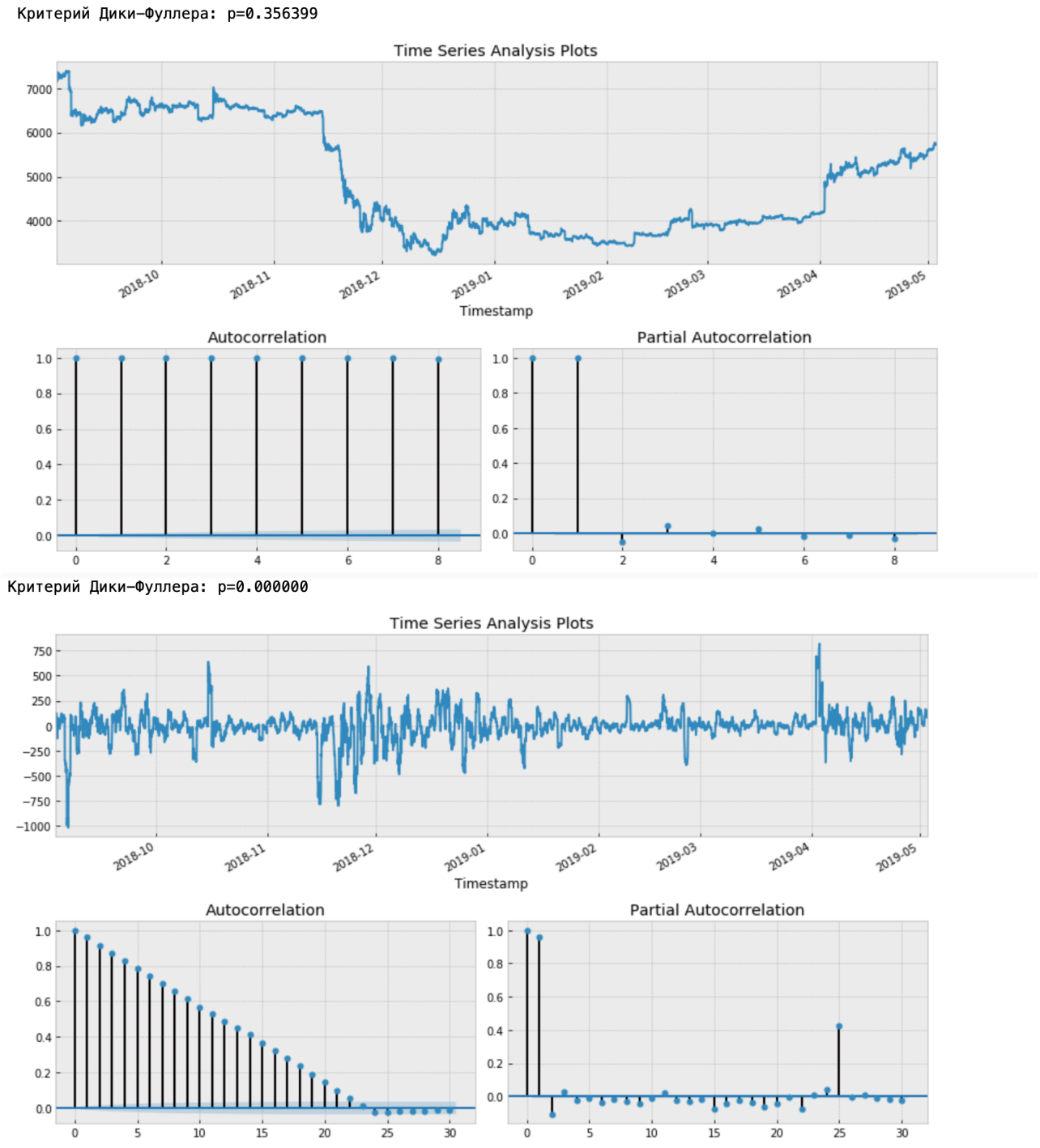

Рисунок 3.6 — Тест Діккі-Фуллера та його АКФ та ЧАКФ

В рамках магістерської роботи було побудовано ряд моделей. Моделі

експоненційного згладжування приймають одну змінну, адже вони працюють суто з часовим рядом. В свою чергу для нейронної мережі архітектури LSTM на її вхід подаються вектори різної розмірності.

3.4 Результати апробації програмного продукту

Для апробації продукту на реальних даних було використано вибірку даних із торгової криптовалютної біржі, що знаходиться у вільному доступі. Датасет містить 300000 записів по транзакціях купівлі продажу валюти, та включає в себе інформацію із трьох змінних по кожній транзакції.

Шляхом генерації нових змінних та групуванню за часовими характеристиками отримали кінцевий набір даних із 387 записами. Розділ вибірки на навчальну та перевірочну проводився у відношенні 70/30%. Враховуючи той факт, що кожна точка відповідає заданому часу дискретизації.

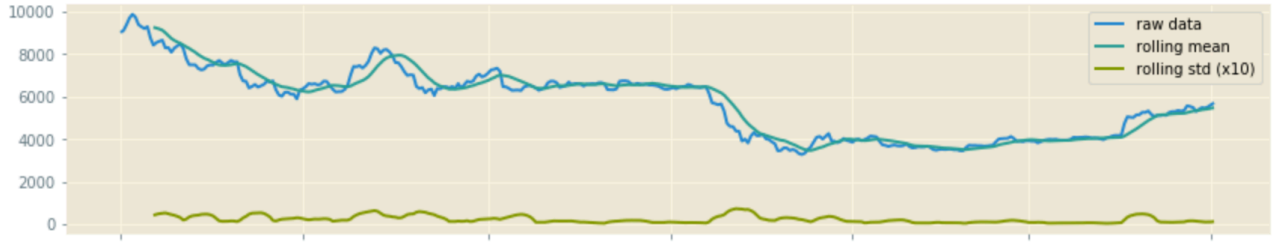

Рисунок 3.7 — Результат вхідного часового ряду із ковзним середнім

Приведемо результати роботи мережі LSTM рис 3.8-3.10.

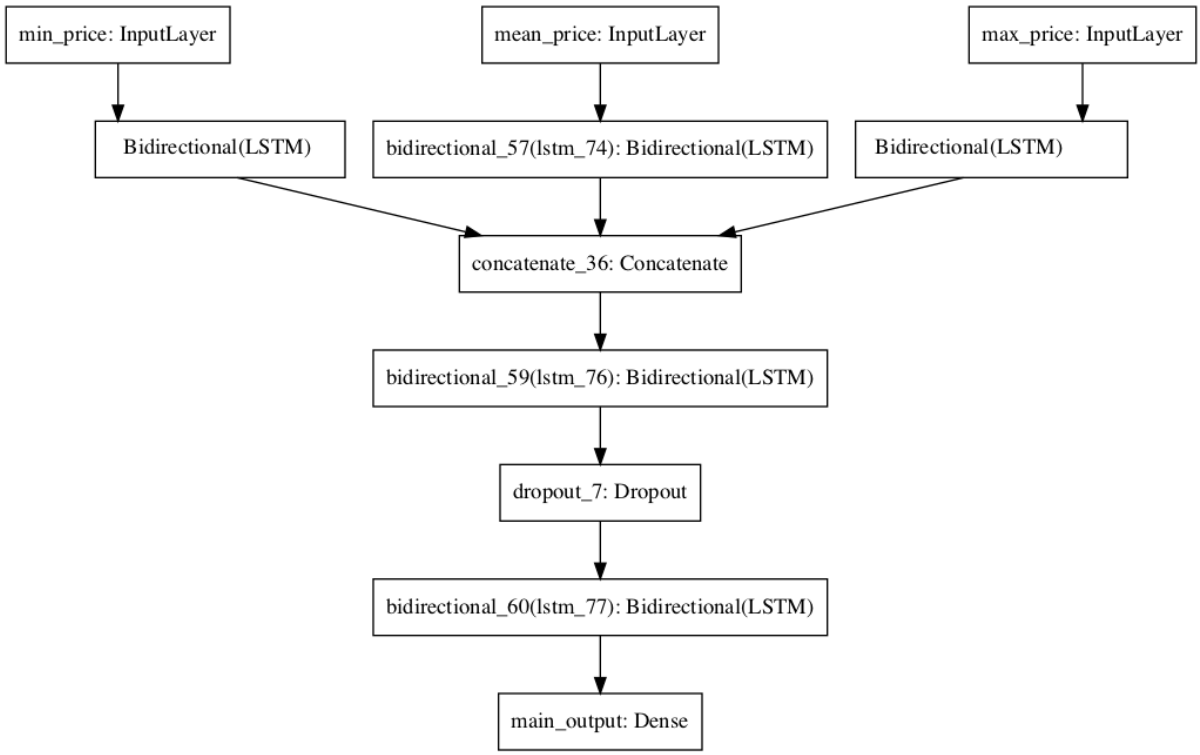

Рисунок 3.8 — Приклад архітектури нейронної мережі LSTM

Навчання нейронної мережі проводилось на 50 епохах та якості економії ресурсів та обчислювальних потужностей було використано колбек. Нижче наведено графік залежності функції похибок від кількості епох для навчальної та перевірочної вибірок.

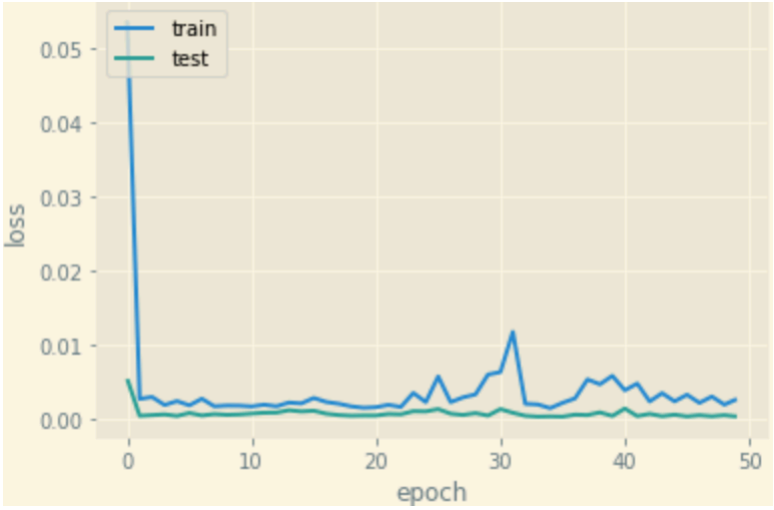

Рисунок 3.9 — Приклад навчання нейронної мережі на 50 епохах із вбудованим колбеком

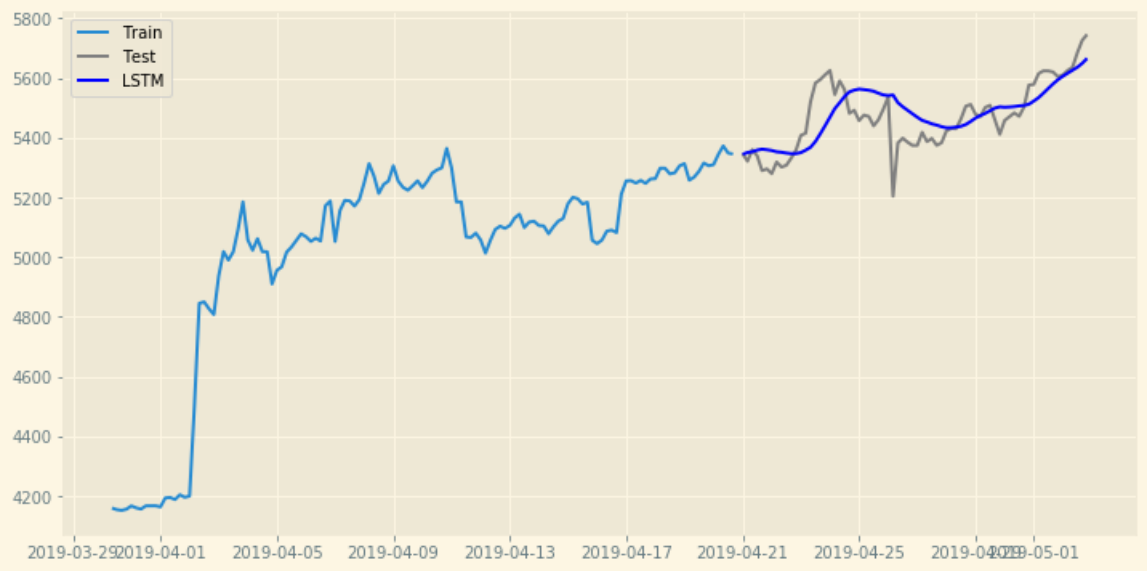

Рисунок 3.10 — Результат роботи LSTM моделі

В результаті роботи нейронної мережі LSTM отримали наступні значення: критерій MAPE приймає значення 1.54%. Для метрики пов'язаної із перевіркою правильності прогнозування тренду маємо наступні значення (табл. 3.1).

Таблиця 3.1 — Confusion matrix для результатів правильності визначення напрямку тренду LSTM

Total	Predicted UP	Predicted DOWN
Actual UP	37	9
Actual DOWN	12	13

Значення критерію $f1_{score} = 0.8$.

Наступною побудованою моделлю була модель Холта-Вінтера, параметри цієї моделі підбиралися на сітці, методом перебору. На рис 3.11 показано оптимально підібрані параметри для моделі Холта-Вінтера.

Dep. Variable:	endog	**No. Observations:**	2106
Model:	ExponentialSmoothing	**SSE**	9534394.733
Optimized:	True	**AIC**	17750.036
Trend:	Multiplicative	**BIC**	17812.214
Seasonal:	Additive	**AICC**	17750.210
Seasonal Periods:	7	**Date:**	Sun, 08 Dec 2019
Box-Cox:	False	**Time:**	22:21:32
Box-Cox Coeff.:	None		

	coeff	code	optimized
smoothing_level	0.9473684	alpha	True
smoothing_slope	0.0526316	beta	True
smoothing_seasonal	0.0526316	gamma	True
initial_level	5710.8669	l.0	True
initial_slope	1.0073550	b.0	True

Рисунок 3.11 — Параметри моделі Холта-Вінтера

Параметри моделі Холта-Вінтера підбирались на дискретній сітці.

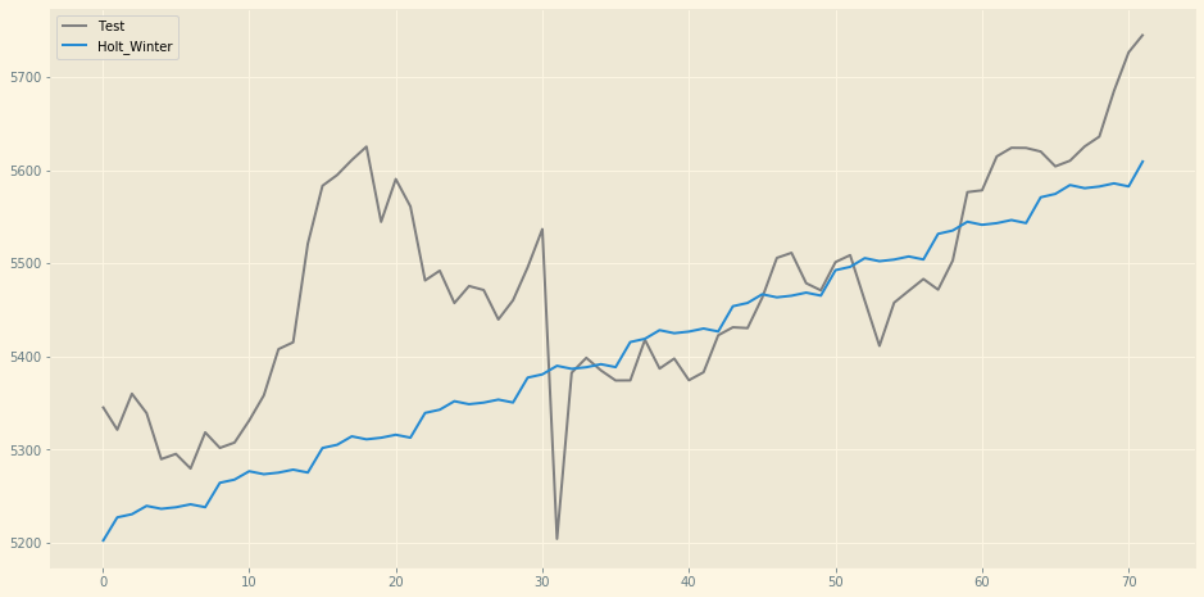

Рисунок 3.12 — Результат прогнозування за моделлю Холта-Вінтера

Для даної моделі отримали наступні значення критеріїв: MAPE = 2.59%.

Таблиця 3.2 — Confusion matrix для результатів правильності визначення напрямку тренду Holt's Exponential Smoothing

Total	Predicted UP	Predicted DOWN
Actual UP	34	19
Actual DOWN	12	6

Значення критерію $f1_{score} = 0.79$.

Порівняємо одержані результати моделей та об'єднаємо дані у зведену результуючу таблицю.

Таблиця 3.3 — Результати роботи моделей

Model	MAPE(%)	TP	TN	FP	FN	f1_score
Simple Exponential Smoothing	2.65	29	20	14	8	0.77
Holt's Exponential Smoothing	2.59	34	19	12	6	0.79
Long Short Term Memory	1.54	30	16	17	6	0.8

Найкращий результат за показником MAPE дала нейронна мережа LSTM. За критерієм прогнозування направленості тренду f1 score кращі результати також дала LSTM мережа. На відміну від методу Холта-Вінтера, LSTM нейронні мережі мають один недолік - це необхідність великих обчислювальних потужностей.

Висновки до розділу 3

В даному розділі спроектовано систему для підтримки та прийняття рішень для прогнозування курсу криптовалют. Реалізована система складається з наступних структурних складових — пристрій вводу-виводу, блок інтерфейсу, блоки завантаження та збереження інформації, блок обробки даних, блок аналізу та прогнозування та також виведення результатів.

Згідно із описаною СППР було реалізовано програмний продук, що дозволяє завантажувати дані, проводити маніпуляції із даними (попередній аналіз та обробку), будуватимоделі та обчислювати характеристики якості побудованої моделі згідно обраних критеріїв, зберігати результати прогнозування.

Визначено мінімальні допустимі технічні характеристики комп'ютера для адекватної роботи розробленого продукту. Описано такі показники як: тактова частота процесору, об'єм оперативної пам'яті, об'єм вільного місця на диску, операційна система, додаткове програмне забезпечення та бібліотеки, без яких неможлива робота ПП, та допоміжні пристрої необхідні для повноцінної роботи оператора.

Реалізовано ПП для роботи із фінансовими часовими рядами за допомогою нейронних мереж LSTM архітектури та моделей експоненційного згладжування; ПП апробовано на реальних даних взятих із відкритих джерел інформації та проведено порівняльний аналіз із обгрунтованим вибором кращої моделі.

РОЗДІЛ 4
РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ

В останні роки набув великої популярності такий вид малого підприємництва як стартап.

Стартап-проект – є комерційним проектом, який знаходиться в стані розробки, або нещодавно вийшов на ринок. Характерною особливістю стартапу, що відрізняє його від малого бізнесу, є оригінальність та інновації, він не може бути копією вже реалізованих ідей. При цьому проект не обов'язково повинен бути масштабного характеру, головне, щоб він був креативним, а його завдання – спрощувати людям будь-які дії в їх повсякденному житті.

Наразі, з появою Інтернету та сучасних технологій, стало простіше заходити на ринок, знаходити інвесторів та споживачів. З'явилося набагато більше можливостей для розвитку свого проекту за кордоном, ніж раніше. Проте розробка стартапу є досить ризикованим завданням. Не всім вдається довести свій стартап-проект до ринкового впровадження. За статистикою успіху досягає лише 10-20% від усіх стартап-проектів.

Запуск стартапу передбачає цілий ряд обов'язкових дій, в межах яких визначають ринкові перспективи стартапу, графік розробки, принципи організації виробництва, заходи з залучення інвесторів та аналіз ризиків.

4.1 Опис ідеї проекту

У таблиці 4.1 подано зміст ідеї стартап-проекту, можливі напрямки застосування та основні вигоди, що може отримати користувач товару. У таблиці 4.2 визначені сильні, слабкі та нейтральні сторони проекту.

Таблиця 4.1 — Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Програмний продукт для прогнозування курсу криптовалют із подальшим прийняттям інтелектуального рішення стосовно торгівлі	Криптовалютні біржі	Дозволяє користувачам з різним рівнем підготовки проводити необхідну попередню обробку даних для побудови прогнозуючої моделі, будувати предиктивну модель та одержувати прогнозні дані на основі побудованої моделі

Таблиця 4.2 — Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/п	Техніко-економічні характеристики ідеї	(потенційні) товари/концепції конкурентів			
		Мій проект	Deductor Credit Scorecard Modeler	IBM SPSS Modeler	SAS Enterprise Miner
1.	Ціна	Низька	Середня	Висока	Висока
2.	Функціонал	Вузький	Вузький	Широкий	Широкий

Отже, з табл. 4.2 можна визначити, що ціна є сильною характеристикою для потенційного товару, а функціонал, зважаючи на напрямки застосування товару, є нейтральною властивістю.

4.2 Технологічний аудит ідеї проекту

За результатами аналізу таблиці 4.3 можна зробити висновок про можливість технологічної реалізації проекту.

Таблиця 4.3 — Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технології
1.	Програмний продукт для прогнозування курсу криптовалют на основі інтелектуального аналізу даних	Прогнозування на основі авторегресійної моделі	Наявна	Доступна
2.		Прогнозування на основі нейронної мережі каскадного типу	Наявна	Доступна
(Обрана технологія реалізації ідеї проекту: прогнозування на основі методу логістичної регресії				

4.3 Аналіз ринкових можливостей запуску стартап-проекту

Визначення ринкових можливостей, які можна використати під час ринкового впровадження проекту, та ринкових загроз, які можуть перешкодити

реалізації проекту, дозволяє спланувати напрями розвитку проекту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів-конкурентів.

Проведемо аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку (табл. 4.4).

Таблиця 4.4 — Попередня характеристика потенційного ринку стартапу

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	3
2	Загальний обсяг продаж, грн/ум.од	100 000 ум.од
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Немає
5	Специфічні вимоги до стандартизації та сертифікації	Немає
6	Середня норма рентабельності в галузі (або по ринку), %	75%

За результатами аналізу таблиці 4.4 можна зробити висновок, що ринок є привабливим для входження за попереднім оцінюванням.

Визначимо потенційні групи клієнтів, їх характеристики, та сформуємо орієнтовний перелік вимог до товару для кожної групи (табл. 4.5).

Таблиця 4.5 — Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія	Відмінності у поведінці груп клієнтів	Вимоги споживачів
1	Прийняття рішення щодо здійснення фінансових операцій стосовно криптовалюти	Трейдингові компанії	Відмінність сфер діяльності клієнтів (торгівля в якості інвестора, трейдингова компанія)	Висока точність прогнозування. Простий у використанні. Швидкодія при обробці значного об’єму інформації

Проведемо аналіз ринкового середовища: таблиці факторів, що сприяють ринковому впровадженню проекту, та факторів, що йому перешкоджають (табл. №№ 4.6-4.7).

Таблиця 4.6 — Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Наявність великої конкуренції	Вихід на ринок великої компанії	Вихід з ринку. Обрати нову цільову аудиторію. Передбачити переваги продукту, щоб повідомити про них саме після виходу великої компанії на ринок
2	Зміна потреб користувачів	Користувачам необхідні рішення з іншим функціоналом	Передбачити можливість додавання нового функціоналу до продукту

Таблиця 4.7 — Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Відсутність конкуренції	Відсутність аналогічних продуктів для користувача на вітчизняному ринку	Локалізація та адаптація сервісу для локальних груп. Адаптація до вітчизняних особливостей
2	Поява нових цільових груп клієнтів	Потреба в аналогічному продукті в інших сферах діяльності	Адаптація продукту під нові сфери використанняу

Проведемо аналіз пропозиції: визначимо загальні риси конкуренції на ринку (табл. 4.8).

Таблиця 4.8 — Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Вказати тип конкуренції - монополістична	Існує декілька фірм-конкурентів	Підтримка якості продукту та постійні вдосконалення
2. За рівнем конкурентної боротьби - інтернаціональний	Фірми конкуренти з різних країн	Підтримувати продукт на національному ринку
3. За галузевою ознакою - внутрішньогалузева	Продукт використовується в одній галузі	Вдосконалювати продукт для застосування в інших галузях
4. Конкуренція за видами товарів: - товарно-родова	Присутня конкуренція з боку товарів-замінників	Розширювати функціонал продукту
5.За характером конкурентних переваг - неціна	Вдосконалення якості продукції, технології виробництва, інновацій	Випускати нові товари, які принципово відрізняються від своїх попередників та представляють модернізований варіант старої моделі
6. За інтенсивністю - немарочна	Роль торгової марки незначна	Приділяти увагу якості продукту а не бренду компанії

Після аналізу конкуренції проведемо більш детальний аналіз умов кон-

куренції в галузі (за моделлю 5 сил М. Портера) (табл. 4.9).

Таблиця 4.9 — Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	Bitcoin Ticker Widget, Cryptolab	APPBTC	Диференціація витрат, розширення каналів збуту	Контроль якості продукту	Наявність більш широкого функціоналу , зручнішого інтерфейсу
Висновки:	Середня конкурентна боротьба з вже існуючими на ринку гравцями	Є можливості виходу на ринок, але є і конкуренти. Строки – пів року	Постачальники не диктують умови роботи	Клієнти диктують умови роботи на ринку	Обмеження для роботи на ринку через товари замінники

На основі аналізу конкуренції (табл. 4.9), а також із урахуванням характеристик ідеї проекту (табл. 4.2), вимог споживачів до товару (табл. 4.5) та факторів маркетингового середовища (табл.№№4.6-4.7) визначимо та обґрунтуємо перелік факторів конкурентоспроможності (табл. 4.10).

Таблиця 4.10 — Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Ціна	Більш доступна ціна збільшує кількість потенційних клієнтів
2	Функціонал	Функціонал направлений на предметну область
3	Зручний інтерфейс	Зручний інтерфейс робить продукт більш привабливим для клієнтів

За визначеними факторами конкурентоспроможності (табл. 4.10) проведемо аналіз сильних та слабких сторін стартап-проекту (табл. 4.11).

Таблиця 4.11 — Порівняльний аналіз сильних та слабких сторін «SmartTrade»

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у Бали порівнянні з “SmartTrade”						
			-3	-2	-1	0	+1	+2	+3
1	Ціна	18		+					
2	Функціонал	10					+		
3	Зручний інтерфейс	12				+			

Складемо SWOT-аналіз (матриця аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities)) (табл. 4.12) на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін (табл. 4.11).

Таблиця 4.12 — SWOT-аналіз стартап-проекту

Сильні сторони: ціна, зручний інтерфейс	Слабкі сторони: функціонал
Можливості: Низька конкуренція, поява нових потреб споживачів	Загрози: Висока конкуренція, не відповідність потребам споживачів

На основі SWOT-аналізу визначимо альтернативи ринкової поведінки (перелік заходів) для виведення стартап-проекту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок (табл. 4.13).

Таблиця 4.13 — Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Створення програмного забезпечення	80%	3 місяці
2	Створення веб-сервісу	60%	5 місяці

4.4 Розроблення ринкової стратегії проекту

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів (табл. 4.14).

Таблиця 4.14 — Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Трейдингові компаніїї	Висока	Високий	Середня	Середня складність
2	Інші фінансові установи	Середня	Середній	Помірна	Висока складність
Які цільові групи обрано: 1					

Для роботи в обраних сегментах ринку сформуємо базову стратегію розвитку (табл. 4.15).

Таблиця 4.15 — Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції	Базова стратегія розвитку*
1	Надання товару важливих з точки зору споживача відмітних властивостей, які роблять товар відмінним від товарів конкурентів	Визначити потреби кожної з цільових груп, розробити стратегії приваблення споживачів та маркетингові комунікації	Оперативне реагування на зміни в ринковому попиті, орієнтованість на кінцевого споживача, висока якість продукту	Стратегія диференціації

Оберемо стратегію конкурентної поведінки (табл. 4.16).

Таблиця 4.16 — Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першо-прохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки*
1	Не є першо-прохідцем	Шукати нових	Ні	Стратегія заняття конкурентної ніші

Сформуємо ринкову позицію, за якою споживачі мають ідентифікувати проект(табл. 4.17).

Таблиця 4.17 — Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту
1	Простий та зручний користувацький інтерфейс, надійність та безпека, швидкість роботи продукти	Стратегія диференціації	Позиція на основі порівняння продукту компанії з продуктами конкурентів. Відмінні особливості споживачів	Автоматизація робочих процесів, зниження кредитних ризиків, зниження навантаження та часу

4.5 Розроблення маркетингової програми стартап-проекту

У табл. 4.18 підсумуємо результати попереднього аналізу конкурентоспроможності товару.

Таблиця 4.18 — Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами
1	Автоматизація робочих процесів	Продукт автоматизує такі процеси, як обробка даних та прийняття рішення щодо здійснення операцій на криптовалютній біржі	Після впровадження продукту процес прийняття рішення щодо купівлі/продажу криптовалюти спрощується
2	Зниження навантаження та часу	Продукт знижує навантаження на люде, відповідальних за прийняття рішень	Персоналу фінансових установ не потрібно самостійно аналізувати великий об’єм даних, що знижує навантаження на прискорює роботу

Розроблена трирівнева маркетингова модель товару(табл. 4.19).

Таблиця 4.19 — Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові
I. Товар за задумом	Програмний продукт для прогнозування кредитоспроможності фізичних осіб. Повинен бути зручним, швидким та безпечним
II. Товар у реальному виконанні	Властивості/характеристики М/Нм Вр/Тх /Тл/Е/Ор
	1. Попередня обробка даних 2. Побудова предиктивних моделей 3. Прогнозування курсу криптовалют
	Якість: проходження тестування
	Пакування: відсутнє
	Марка: “SmartTrade ”
III. Товар із підкріпленням	До продажу: відсутнє
	Після продажу: навчання персоналу, супровід, технічна підтримка
Вихідний код програмного продукту є закритим, та не передається клієнтам і третім особам.	

Визначимо цінові межі, якими необхідно керуватись при встановленні ціни на товар (табл. 4.20).

Таблиця 4.20 — Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/-послугу
1	2500$	2000$	Високий рівень доходів	Базова покупка та впровадження: нижня межа - 1000$, верхня межа - 2000$.

Визначимо оптимальну систему збуту (табл. 4.21).

Таблиця 4.21 — Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товар	Глибина каналу збуту	Оптимальна система збуту
1	Цільові клієнти – банківські установи, які бажають впровадити у своїй роботі сучасні засоби,допоможуть автоматизувати робочі процеси. Вони цікавляться інноваційними рішеннями, відвідують тематичні семінари та конференції	Формування попиту і стимулювання збуту. Встановлення контактів із споживачами. Просування маркетингової інформації	Нульова або однорівнева (сервіс безпосередньо продається споживачам та через посередників)	Прямий канал збуту до споживача, мінімізувати витрати на додаткові канали збуту

Розроблена концепція маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів (табл. 4.22).

Таблиця 4.22 — Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Цільові клієнти - фінансові установи, трейдингові компанії та приватні особи, що займаються купівлею продажу криптовалют на ринку. Вони цікавляться інноваційними рішеннями, відвідують тематичні семінари та конференції	Конференції, форуми, новини у сфері інноваційних технологій, періодичні видання у професійних галузях	Позиція на основі порівняння продукту компанії з продуктами конкурентів. Відмінні особливості споживачів	- інформувати про новий продукт та його переваги; - сформувати сприятливу думку; - сформувати образ марки та її виробника у свідомості споживачів; - збільшити потік покупців	Зменшуємо фінансові ризики. Прискорюємо та автоматизуємо процес прийняття рішення

Висновки до розділу 4

В даному розділі проведено аналіз створення та виведення на ринок стартап-проекту на основі програмного продукту, який було розроблено в рамках магістерської дисертації. В межах цього аналізу було розроблено опис самої ідеї проекту, визначено загальні напрями використання товару, проаналізовано ринкові можливості щодо впровадження проекту, визначено відмінності від конкурентів та розроблено стратегію виходу на ринок. Узагальнюючи проведений аналіз, можна зазначити, що є можливість ринкової комерціалізації проекту. Наявний попит, динаміка ринку зростає. З огляду на потенційні групи клієнтів, а саме фінансові установи, та високий рівень конкурентоспроможності проекту, є достатні перспективи для впровадження стартапу. Отже, подальша імплементація проекту є доцільною.

ВИСНОВКИ

Дана робота присвячена аналізу, побудові та використанню прогнозуючих моделей для спрощення прийняття рішень при торгівлі на фінансових криптовалютних біржах.

Після ознайомлення з теоретичним матеріалом щодо суті проблематики та принципів роботи технології блокчейн, основними статистичними та математичними методами побудови предиктивних моделей для прогнозування курсу криптовалют, було побудовано СППР для прийняття рішень щодо операцій із криптовалютними парами в фінансових установах.

В якості практичного прикладу застосування СППР, було розроблено програмний продукт з використанням технологій Python у середовищі розробки Jupyter Notebook. У даній системі було реалізовано нейронну мережу для короткострокового прогнозування курсу криптовалют. Для знаходження оцінок параметрів моделі було використано метод максимальної правдоподібності з використанням методу градієнтного спуску.

Отримані результати порівняння статистичних характеристик якості побудованих прогнозуючих моделей показали, що одержана нейронна мережа, реалізована у розробленому програмному продукті, дає не гірші результати ніж інші методи, реалізовані у комерційних аналогах.

Результати магістерської дисертації:

а) запропоновано архітектуру системи підтримки прийняття рішень для прогнозування курсу криптовалют;
б) розроблено програмний продукт для аналізу та обробки даних, побудови регресійної моделі на основі нейронних мереж для прогнозування курсу крипотвалют;
в) розроблений ПП апробовано на вибірці з 30000 криптовалютних операцій на криптовалютній біржі Poloniex;
г) виконано порівняльний аналіз з іншими методами прогнозування, реалізованими в комерційних системах.

Подальшими напрямками роботи можуть бути питання, що стосуються:

а) вдосконалення розробленої архітектури нейронної мережі;
б) реалізації методів для автоматизації процесу торгівлі із використанням можливостей біржі Poloniex через API.

Розроблений програмний продукт показав прийнятні результати, що підтверджує раціональність використання обраного методу.

ПЕРЕЛІК ПОСИЛАНЬ

1. Николенко, С., Кадурин, А., & Архангельская, Е. *Глубокое обучение*, 2017, СпБ: Питер, 432 с.

2. Bain A. *The Senses and the Intellect*, London: Parker, 1855. 215 p.

3. Auer P., Cesa-Bianchi N., Fischer P. *Finite-Time Analysis of the Multiarmed Bandit Problem*, Machine Learning, 2002, vol. 47, no. 2–3. P. 235–256.

4. Tesauro G. *Practical Issues in Temporal Difference Learning*, Machine Learning, 1992, vol. 8. P. 257–277.

5. Thorndike E. L. *Animal Intelligence: An Experimental Study of the Associative Processes in Animals*, New York: Macmillan, 1898. 368 p.

6. Statistical Analysis System – Режим доступу:
sas.com/ru_ua/home.html
Дата доступу: 08.09.2019

7. Shannon C. *This Mouse Is Smarter Than You Are*, Popular Science, 1952. P. 99–101.

8. Minsky M. *Neural Nets and the Brain Model Problem* Princeton: Princeton University, 1954. 157 p.

9. Goodfellow, I., Bengio, Y., & Courville, A. *Deep learning*. MIT press., 511 p.

10. Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. Psychological review, 65(6), 386.

11. Andrej Karpathy. The unreasonable effectiveness of recurrent neural networks. 2015. URL: http://karpathy.github.io/2015/05/21/rnn-effectiveness/.

12. Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. (2017). LSTM: A search space odyssey. IEEE transactions on neural networks and learning systems, 28(10), 2222-2232.

13. Jones E, Oliphant E, Peterson P, et al. SciPy: Open Source Scientific Tools for Python, 2001-, http://www.scipy.org/ [Online; accessed 2018-06-12].

14. Chollet, François and others. Keras. 2015. URL: https://keras.io.

15. Andrej Karpathy. The unreasonable effectiveness of recurrent neural networks. 2015. URL: http://karpathy.github.io/2015/05/21/rnn-effectiveness/.

16. Stationarity in time series. – Режим доступу: https://towardsdatascience.com/stationarity-in-time-series-analysis-90c94f27322

17. Howard R. A. *Dynamic Programming and Markov Processes*, Cambridge, MA: MIT Press, 1960. 231 p.

18. Моделі нейронних мереж – Режим доступу: https://studme.com.ua/1246122010028/neural/models.htm Дата доступу: 27.08.2019

19. Архітектура нейронних мереж. – Режим доступу: http://techn.sstu.ru/kafedri/подразделения/1/MetMat/Terin/neiro/neiro.htm Дата доступу: 28.09.2019

20. Simonyan K. Very Deep Convolutional Networks for Large-Scal Image Recognition – Режим доступу до ресурсу: http://arxiv.org/abs/1409.1556 Дата доступу: 08.05.2019

21. Лисе А.А., Степанов М.В. Нейронные сети и нейрокомпьютеры: учеб. пособие. ГЭТУ. – СПб. 2009. 64 с.

22. Seasonal ARIMA with Python. - Режим доступу: https://www.seanabu.com/2016/03/22/time-series-seasonal-ARIMA-model-in-python/

23. Distribution of the Estimators for Autoregressive Time Series with a Unit Root. - Режим доступу: https://www.jstor.org/stable/2286348?seq=1

24. Spyros Makridakis, Steven C. Wheelwrigh, Forecasting methods for management, Fifth Edition; Wiley, New York, 1989, pp. 470.

25. Additive and Multiplicative models. - Режим доступу: http://www-ist.massey.ac.nz/dstirlin/CAST/CAST/Hmultiplicative/multiplicative1.html

26. Моделі нейронних мереж. – Режим доступу: https://www.intuit.ru/studies/courses/57/57/lecture/1682?page=3

27. Money blockchains and social scalability. – Режим доступу: https://bitnovosti.com/2017/04/17/ money-blockchains-and-social-scalability-part4/comment-page-1/

28. Алгоритмы / Хэш-функция SHA-256 - Режим доступу:

https://medium.com/dtechlog/алгоритмы-хэш-функция-sha-256-9862302f942f

29. Design a Data and Analytics Strategy - Режим доступу: https://www.gartner.com publications/data-analytics-strategy Дата доступу: 08.10.2019

30. Бідюк П.І., Коршевнюк Л.О. Проектування комп'ютерних інформаційних систем підтримки прийняття рішень: Навчальний посібник. — Київ: ННК „ІПСА" НТУУ „КПІ", 2010. — 340 с.

31. Бідюк П.І. Часові ряди: моделювання та прогнозування / Бідюк П.І., Савенков О.І. Баклан І.В. – Київ: ЕКМО, 2004. – 144 с.

ДОДАТОК А ЛІСТИНГ ПРОГРАМИ

```
#!/usr/bin/env python
# coding: utf-8

# # Importing necessary libs

# In[1]:


import pandas as pd
import glob
import numpy as np
import datetime


# In[2]:


import warnings
warnings.filterwarnings("ignore")


# # Reading CSVs

# In[3]:


path = r'DATA/BTCUSD_20180101_20190502/'
all_files = glob.glob(path + "/*.csv")

#converting time from milliseconds to standart format
convert_mask = lambda x: datetime.datetime.fromtimestamp(float(x) / 1e3)

#colum names for reading
column_names = ['Timestamp', 'Amount', 'Price']

li = []

for filename in all_files:
frame = pd.read_csv(filename, header=0,
usecols=column_names, index_col=None,
parse_dates=['Timestamp'], date_parser=convert_mask)
li.append(frame)

df = pd.concat(li, axis=0, ignore_index=True)


# # Reducing memory usage

# In[4]:


df.head(20)
```

```
# In[5]:


df.info()


# In[6]:


def reduce_mem_usage(props):
start_mem_usg = props.memory_usage().sum() / 1024**2
print("Memory usage of properties dataframe is :",start_mem_usg," MB")
for col in props.columns:
if props[col].dtype != '<M8[ns]':  # Exclude Datetime

# Print current column type
print("******************************")
print("Column: ",col)
print("dtype before: ",props[col].dtype)

# make variables for Int, max and min
IsInt = False
mx = props[col].max()
mn = props[col].min()

# Integer does not support NA, therefore, NA needs to be filled
if not np.isfinite(props[col]).all():
NAlist.append(col)
props[col].fillna(mn-1,inplace=True)

# test if column can be converted to an integer
asint = props[col].fillna(0).astype(np.int64)
result = (props[col] - asint)
result = result.sum()
if result > -0.001 and result < 0.001:
IsInt = True


# Make Integer/unsigned Integer datatypes
if IsInt:
if mn >= 0:
if mx < 255:
props[col] = props[col].astype(np.uint8)
elif mx < 65535:
props[col] = props[col].astype(np.uint16)
elif mx < 4294967295:
props[col] = props[col].astype(np.uint32)
else:
props[col] = props[col].astype(np.uint64)
else:
if mn > np.iinfo(np.int8).min and mx < np.iinfo(np.int8).max:
props[col] = props[col].astype(np.int8)
elif mn > np.iinfo(np.int16).min and mx < np.iinfo(np.int16).max:
props[col] = props[col].astype(np.int16)
elif mn > np.iinfo(np.int32).min and mx < np.iinfo(np.int32).max:
props[col] = props[col].astype(np.int32)
```

```
elif mn > np.iinfo(np.int64).min and mx < np.iinfo(np.int64).max:
props[col] = props[col].astype(np.int64)

# Make float datatypes 32 bit
else:
props[col] = props[col].astype(np.float32)

# Print new column type
print("dtype after: ",props[col].dtype)
print("*******************************")

# Print final result
print("___MEMORY USAGE AFTER COMPLETION:___")
mem_usg = props.memory_usage().sum() / 1024**2
print("Memory usage is: ", mem_usg, " MB")
print("This is ",100 * mem_usg / start_mem_usg,"% of the initial size")
return props


# # Data preparing

# In[7]:


#df['Datetime'] = pd.to_datetime(df.Timestamp, unit='ms')

df.sort_values(by=['Timestamp'], inplace=True)


# In[8]:


df = reduce_mem_usage(df)


# # FE

# In[9]:


def fe(df, start_date, stop_date, freq = 3600):

temp = replace_price(df)

#filters
f1 = temp['Timestamp'] > start_date
f2 = temp['Timestamp'] < stop_date
#f3 = df['Amount'] > 0 if order_type > 0 else df['Amount'] < 0

temp = temp[f1 & f2]

result = temp.groupby([pd.Grouper(key = 'Timestamp', freq = str(freq) + 's')]).agg({
'Price': {
'Min Buy price': lambda x: x[x > 0].min(),
'Max Buy price':  lambda x: x[x > 0].max(),
'Mean Buy price': lambda x: x[x > 0].mean(),
'Min Sell price': lambda x: x[x < 0].min(),
```

```
'Max Sell price':  lambda x: x[x < 0].max(),
'Mean Sell price': lambda x: x[x < 0].mean(),

},
'Amount': {
'Sum Buy Amount': lambda x: x[x > 0].sum(),
'Sum Sell Amount': lambda x: x[x < 0].sum(),
'Total Count': lambda x: x.count(),
'Order Buy Count': lambda x: x[x > 0].count(),
'Order Sell Count': lambda x: x[x < 0].count(),
}
})

#drop 2storey indexes
result = result.droplevel(0, axis=1)

return result

def fe_mean(df, start_date, stop_date, freq = 3600):

temp = replace_price(df)

#filters
f1 = temp['Timestamp'] > start_date
f2 = temp['Timestamp'] < stop_date
#f3 = df['Amount'] > 0 if order_type > 0 else df['Amount'] < 0

temp = temp[f1 & f2]

result = temp.groupby([pd.Grouper(key = 'Timestamp', freq = str(freq) + 's')]).agg({
'Price': {
'Mean Buy price': lambda x: x[x > 0].mean(),

}
})

#drop 2storey indexes
result = result.droplevel(0, axis=1)

return result

def replace_price(df):

df['Price'][df.Amount < 0] = - df['Price']
return df


def create_lag(temp, vars_list, n_lags):

for var in vars_list:
for i in range(1, n_lags):
temp[var + ' ' + str(i)] = temp[var].shift(i)

return temp


# In[14]:
```

```
l = fe(df, '2018-01-01', '2019-05-03', 3600)
l

# In[10]:


l2 = fe_mean(df, '2018-05-01', '2019-05-03', 24 * 3600)


# In[11]:


# # ABT table

# In[18]:


result = create_lag(l, ['Min Buy price', 'Mean Buy price'], 10)
result


# # Plot

# In[19]:


#Testing plot abilities


# In[20]:


from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot
from plotly import graph_objs as go
init_notebook_mode(connected = True)

def plotly_df(df, title = ''):
data = []

trace = go.Scatter(
x = df.index,
y = df['Min Buy price'],
mode = 'lines',
name = 'Min Buy price'
)
data.append(trace)

'''
for column in df.columns:
trace = go.Scatter(
x = df.index,
y = df[column],
mode = 'lines',
name = column
)
```

```
data.append(trace)
'''
layout = dict(title = title)
fig = dict(data = data, layout = layout)
iplot(fig, show_link=False)

plotly_df(l, title = "BTC")

# # Fuller Test

# In[22]:


l['Mean Buy price'].dropna(inplace=True)


# In[23]:


def tsplot(y, lags=None, figsize=(12, 7), style='bmh'):
if not isinstance(y, pd.Series):
y = pd.Series(y)
with plt.style.context(style):
fig = plt.figure(figsize=figsize)
layout = (2, 2)
ts_ax = plt.subplot2grid(layout, (0, 0), colspan=2)
acf_ax = plt.subplot2grid(layout, (1, 0))
pacf_ax = plt.subplot2grid(layout, (1, 1))

y.plot(ax=ts_ax)
ts_ax.set_title('Time Series Analysis Plots')
smt.graphics.plot_acf(y, lags=lags, ax=acf_ax, alpha=0.5)
smt.graphics.plot_pacf(y, lags=lags, ax=pacf_ax, alpha=0.5)

printКритерий(" ДикиФуллера−: p=%f" % sm.tsa.stattools.adfuller(y)[1])

plt.tight_layout()
return

tsplot(l['Mean Buy price'], lags=8)

# In[59]:

# # LSTM

# In[75]:


from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()

close_price = l2['Mean Buy price'].values.reshape(−1, 1)

scaled_close = scaler.fit_transform(close_price)
```

```
# In[94]:

scaled_close = scaled_close[~np.isnan(scaled_close)]
scaled_close = scaled_close.reshape(-1, 1)


# In[116]:


SEQ_LEN = 12

def to_sequences(data, seq_len):
d = []

for index in range(len(data) - seq_len):
d.append(data[index: index + seq_len])

return np.array(d)

def preprocess(data_raw, seq_len, train_split):

data = to_sequences(data_raw, seq_len)

num_train = int(train_split * data.shape[0])

X_train = data[:num_train, :-1, :]
y_train = data[:num_train, -1, :]

X_test = data[num_train:, :-1, :]
y_test = data[num_train:, -1, :]

return X_train, y_train, X_test, y_test


X_train, y_train, X_test, y_test = preprocess(scaled_close, SEQ_LEN, train_split = 0.967)

from tensorflow.keras.layers import Bidirectional, Dropout, Activation, Dense, LSTM
from tensorflow.python.keras.layers import CuDNNLSTM
from tensorflow.keras.models import Sequential


# In[120]:


from keras.layers import Bidirectional, Dropout, Activation, Dense, LSTM
from keras.models import Sequential


# In[121]:


DROPOUT = 0.2
WINDOW_SIZE = SEQ_LEN - 1

model = Sequential()

model.add(Bidirectional(
```

```
LSTM(WINDOW_SIZE, return_sequences=True),
input_shape=(WINDOW_SIZE, X_train.shape[-1])
))
model.add(Dropout(rate=DROPOUT))

model.add(Bidirectional(
LSTM((WINDOW_SIZE * 2), return_sequences=True)
))
model.add(Dropout(rate=DROPOUT))

model.add(Bidirectional(
LSTM(WINDOW_SIZE, return_sequences=False)
))

model.add(Dense(units=1))

model.add(Activation('linear'))


# In[122]:


BATCH_SIZE = 64

model.compile(
loss='mean_squared_error',
optimizer='adam'
)


# In[123]:


history = model.fit(
X_train,
y_train,
epochs=50,
batch_size=BATCH_SIZE,
shuffle=False,
validation_split=0.1
)


# In[124]:


y_hat = model.predict(X_test)


# In[125]:


y_test_inverse = scaler.inverse_transform(y_test)
y_hat_inverse = scaler.inverse_transform(y_hat)


# In[126]:
```

```
def mean_absolute_percentage_error(y_true, y_pred):
y_true, y_pred = np.array(y_true), np.array(y_pred)
return np.mean(np.abs((y_true - y_pred) / y_true)) * 100


# In[131]:


mean_absolute_percentage_error(y_test_inverse, y_hat_inverse)


# In[128]:


plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()


# In[136]:


plt.plot(y_test_inverse, label="Actual Price", color='green')
plt.plot(y_hat_inverse+100, label="Predicted Price", color='red')

plt.title('Bitcoin price prediction')
plt.xlabel('Time [days]')
plt.ylabel('Price')
plt.legend(loc='best')

plt.show();


# In[137]:


mean_absolute_percentage_error(y_test_inverse, y_hat_inverse+100)


# In[130]:


plt.plot(y_train, label="Actual Price", color='green')
#plt.plot(y_hat_inverse, label="Predicted Price", color='red')

plt.title('Bitcoin price prediction')
plt.xlabel('Time [days]')
plt.ylabel('Price')
plt.legend(loc='best')

plt.show();
```

```
# In[14]:


#import requests
import pandas as pd
import json
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
get_ipython().run_line_magic('matplotlib', 'inline')
plt.style.use('Solarize_Light2')

df = 12

train = df.iloc[:-10, :]
test = df.iloc[-10:, :]
pred = test.copy()
df.plot(figsize=(12,3));
plt.title('title');

df['z_data'] = (df['Mean Buy price'] - df['Mean Buy price'].rolling(window=12).mean()) / df['Mean Buy price
df['zp_data'] = df['z_data'] - df['z_data'].shift(12)


# In[15]:


def plot_rolling(df):
fig, ax = plt.subplots(3,figsize=(12, 9))
ax[0].plot(df.index, df['Mean Buy price'], label='raw data')
ax[0].plot(df['Mean Buy price'].rolling(window=12).mean(), label="rolling mean");
ax[0].plot(df['Mean Buy price'].rolling(window=12).std(), label="rolling std (x10)");
ax[0].legend()

ax[1].plot(df.index, df.z_data, label="de-trended data")
ax[1].plot(df.z_data.rolling(window=12).mean(), label="rolling mean");
ax[1].plot(df.z_data.rolling(window=12).std(), label="rolling std (x10)");
ax[1].legend()

ax[2].plot(df.index, df.zp_data, label="12 lag differenced de-trended data")
ax[2].plot(df.zp_data.rolling(window=12).mean(), label="rolling mean");
ax[2].plot(df.zp_data.rolling(window=12).std(), label="rolling std (x10)");
ax[2].legend()

plt.tight_layout()
fig.autofmt_xdate()


# In[16]:


plot_rolling(df)


# In[17]:

from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
```

```
fig, ax = plt.subplots(2, figsize=(12,6))
ax[0] = plot_acf(df.z_data.dropna(), ax=ax[0], lags=20)
ax[1] = plot_pacf(df.z_data.dropna(), ax=ax[1], lags=20)


# In[138]:


#import requests
import pandas as pd
import json
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
from statsmodels.tsa.holtwinters import SimpleExpSmoothing, Holt
import numpy as np
get_ipython().run_line_magic('matplotlib', 'inline')
plt.style.use('Solarize_Light2')


df = l2
train = df.iloc[100:-10, :]
test = df.iloc[-10:, :]
train.index = pd.to_datetime(train.index)
test.index = pd.to_datetime(test.index)
pred = test.copy()

model = SimpleExpSmoothing(np.asarray(train['Mean Buy price']))
model._index = pd.to_datetime(train.index)

fit1 = model.fit()
pred1 = fit1.forecast(10)
fit2 = model.fit(smoothing_level=.2)
pred2 = fit2.forecast(10)
fit3 = model.fit(smoothing_level=.5)
pred3 = fit3.forecast(10)

fig, ax = plt.subplots(figsize=(12, 6))
ax.plot(train.index[150:], train.values[150:])
ax.plot(test.index, test.values, color="gray")
for p, f, c in zip((pred1, pred2, pred3),(fit1, fit2, fit3),('#ff7823','#3c763d','c')):
ax.plot(train.index[150:], f.fittedvalues[150:], color=c)
ax.plot(test.index, p, label="alpha="+str(f.params['smoothing_level'])[:3], color=c)
plt.title("Simple Exponential Smoothing")
plt.legend();

print(mean_absolute_percentage_error(y_test_inverse, pred3))

model = Holt(np.asarray(train['Mean Buy price']))
model._index = pd.to_datetime(train.index)

fit1 = model.fit(smoothing_level=.3, smoothing_slope=.05)
pred1 = fit1.forecast(10)
fit2 = model.fit(optimized=True)
pred2 = fit2.forecast(10)
fit3 = model.fit(smoothing_level=.3, smoothing_slope=.2)
pred3 = fit3.forecast(10)
```

```
fig, ax = plt.subplots(figsize=(12, 6))
ax.plot(train.index[150:], train.values[150:])
ax.plot(test.index, test.values, color="gray")
for p, f, c in zip((pred1, pred2, pred3),(fit1, fit2, fit3),('#ff7823','#3c763d','c')):
ax.plot(train.index[150:], f.fittedvalues[150:], color=c)
ax.plot(test.index, p, label="alpha="+str(f.params['smoothing_level'])[:4]+", beta="+str(f.params['smoothin
plt.title("Holt's Exponential Smoothing")
plt.legend();


# In[134]:


mean_absolute_percentage_error(y_test_inverse, pred3)


# In[23]:


from statsmodels.tsa.holtwinters import ExponentialSmoothing
import numpy as np

model = ExponentialSmoothing(np.asarray(train['Mean Buy price']), trend='mul', seasonal=None)
model2 = ExponentialSmoothing(np.asarray(train['Mean Buy price']), trend='mul', seasonal=None, damped=True)
model._index = pd.to_datetime(train.index)

fit1 = model.fit()
fit2 = model2.fit()
pred1 = fit1.forecast(10)
pred2 = fit2.forecast(10)

fig, ax = plt.subplots(2, figsize=(12, 12))
ax[0].plot(train.index[150:], train.values[150:])
ax[0].plot(test.index, test.values, color="gray", label="truth")
ax[1].plot(train.index[150:], train.values[150:])
ax[1].plot(test.index, test.values, color="gray", label="truth")
for p, f, c in zip((pred1, pred2),(fit1, fit2),('#ff7823','#3c763d')):
ax[0].plot(train.index[150:], f.fittedvalues[150:], color=c)
ax[1].plot(train.index[150:], f.fittedvalues[150:], color=c)
ax[0].plot(test.index, p, label="alpha="+str(f.params['smoothing_level'])[:4]+", beta="+str(f.params['smoo
ax[1].plot(test.index, p, label="alpha="+str(f.params['smoothing_level'])[:4]+", beta="+str(f.params['smoo
ax[0].set_title("Damped Exponential Smoothing");
ax[1].set_title("Damped Exponential Smoothing – zoomed");
plt.legend();


# In[135]:


mean_absolute_percentage_error(y_test_inverse, pred1)


# # ARIMA

# In[26]:
```

```
fig, ax = plt.subplots(2, sharex=True, figsize=(12,6))
ax[0].plot(df['Mean Buy price'].values);
ax[0].set_title("Raw data");
ax[1].plot(np.log(df['Mean Buy price'].values));
ax[1].set_title("Logged data (deflated)");
ax[1].set_ylim(0, 15);

fig, ax = plt.subplots(2, 2, figsize=(12,6))
first_diff = (np.log(df['Mean Buy price'])- np.log(df['Mean Buy price']).shift()).dropna()
ax[0, 0] = plot_acf(np.log(df['Mean Buy price']), ax=ax[0, 0], lags=20, title="ACF - Logged data")
ax[1, 0] = plot_pacf(np.log(df['Mean Buy price']), ax=ax[1, 0], lags=20, title="PACF - Logged data")
ax[0, 1] = plot_acf(first_diff , ax=ax[0, 1], lags=20, title="ACF - Differenced Logged data")
ax[1, 1] = plot_pacf(first_diff, ax=ax[1, 1], lags=20, title="PACF - Differenced Logged data")


# In[29]:


from statsmodels.tsa.stattools import kpss

print(" > Is the data stationary ?")
dftest = kpss(np.log(df['Mean Buy price']), 'ct')
print("Test statistic = {:.3f}".format(dftest[0]))
print("P-value = {:.3f}".format(dftest[1]))
print("Critical values :")
for k, v in dftest[3].items():
print("\t{}: {}".format(k, v))



# In[47]:


from statsmodels.tsa.arima_model import ARIMA

model = ARIMA(np.log(df['Mean Buy price']).dropna()[:-12], (1, 1, 1))
res_111 = model.fit()


# In[51]:


fig, ax = plt.subplots(figsize=(12, 6))
#df.index = pd.to_datetime(df.index, format="%Y-%m")
np.log(df['Mean Buy price']).dropna()[250:].plot(ax=ax);
ax.vlines('2019-04-20', 8, 9, linestyle='--', color='r', label='Start of forecast');

# - NOTE from the official documentation :
# -- The dynamic keyword affects in-sample prediction.
# -- If dynamic is False, then the in-sample lagged values are used for prediction.
# -- If dynamic is True, then in-sample forecasts are used in place of lagged dependent variables.
ax = res_111.plot_predict('2019-04-20', '2019-05-02', dynamic=True, plot_insample=False, ax=ax);


# In[53]:
```

```
res_111.predict('2019-04-20', '2019-05-02',dynamic=True)


# In[64]:


df['Mean Buy price'][:355]


# In[66]:


#building the model
from pmdarima.arima import auto_arima
model = auto_arima(df['Mean Buy price'][:355], trace=True, error_action='ignore', suppress_warnings=True)
model.fit(df['Mean Buy price'][:355])


# In[69]:


forecast = model.predict(n_periods=12)
forecast = pd.DataFrame(forecast,index = df['Mean Buy price'][355:].index,columns=['Prediction'])


# In[72]:


#plot the predictions for validation set
#plt.plot(train, label='Train')
plt.plot(df['Mean Buy price'][355:], label='Real')
plt.plot(forecast, label='Prediction')
plt.show()


# In[ ]:
```